

**UNIVERSIDAD LUTERANA SALVADOREÑA
FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA
LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN.**



Materia: Algoritmos II.

Docente: Licenciado Pedro Antonio Trejo Noble.

Tema Desarrollado: Reto Algoritmo

Alumno/@s

N°	Apellido y Nombres	Carnet
1	Henry David Aguilar Sánchez	AS01134801
3	David Armando Cornejo Alfonso	CA01134754

San Salvador, 1 de junio de 2018

INDICE

Contenido

Contenido.....	4
INTRODUCCIÓN.....	3
OBJETIVOS:	4
Generales:.....	4
Específicos:	4
PLANTEAMIENTO DE PROBLEMA	5
MARCO REFERENCIAL.....	7
Modelo.....	8
Controlador.....	8
Vistas.....	8
MARCO CONCEPTUAL	11
Reto algoritmo.....	11
Diagrama de Flujo	16
Diseño de la base de datos	16
Código	17
CRONOGRAMA DE ACTIVIDADES	20
CONCLUSIONES.....	21
BIBLIOGRAFÍA	22

INTRODUCCIÓN.

A lo largo de toda la vida se implementan algoritmos con el fin de resolver problemas, A esto nace la idea de desarrollar un juego que permita al usuario aprender más de algoritmos.

El objeto principal del juego es que el usuario aprenda de algoritmos de una manera llamativa, saliendo un poco del esquema de aprendizaje que usualmente se implementa.

El juego consistirá en mostrar una serie de preguntas en pantalla, el usuario debe elegir la respuesta correcta, si esto se cumple el usuario avanzara de nivel con preguntas más complejas hasta completar todos los niveles.

Para la implementación del juego se utilizara el lenguaje de programación PHP, este abre muchas posibilidades para el desarrollo de este tipo de juegos, para la interfaz gráfica se hará uso del framework Bootstrap, en cuanto al manejo de la base de datos se hará uso de MySql

OBJETIVOS:

Generales:

- Implementar un juego que permita conocer de algoritmos mediante un lenguaje de programación, base de datos y un framework.

Específicos:

- Mostrar a través del juego una manera interactiva para conocer conceptos acerca de algoritmos.
- Desarrollar un juego mediante el lenguaje de programación PHP
- Desarrollar una base de datos haciendo uso de Mysql
- Desarrollar interfaz gráfica mediante la librería Bootstrap

PLANTEAMIENTO DE PROBLEMA

Los algoritmos que son utilizados durante el desarrollo de una actividad de los humanos, mediante la lógica de la ejecución de una actividad o acción en procesos ordenados.

Se considera de importancia tener conocimiento acerca de la importancia de los algoritmos en la vida cotidiana del ser humano, como se aplican, en qué momento poder hacer uso de ellos, Etc. Por ejemplo la acción de cambiar un neumático desinflado, conlleva a una acción ordenada para tener como resultado un neumático en buenas condiciones.

Que ocurriría si a la hora de resolver un problema no lo hacemos en orden lógico o de manera ordenada. El efecto a obtener sería la resolución de un problema de manera equivocada o la no solución de un problema.

El pretender crear un programa lúdico enfocado a la enseñanza de una manera interactiva y de fácil comprensión para el ser humano, como podemos observar la cantidad de programas de las categorías más solicitadas nivel mundial; como los videojuegos de guerra, deporte, autos entre otros, por lo consiguiente el algoritmo tiene como objetivo mejorar el conocimiento previo de un estudiante de la carrera de computación, sin dejar a un lado a las personas que por primera vez podrían relacionarse con la información presentada. El objetivo del juego es enseñar conceptos básicos de algoritmo para aquellas personas que estén interesadas en aprender y para aquellas que ya tengan una base tratar la manera de poner ese conocimiento a prueba.

En específico se busca fortalecer las capacidades mínimas en la persona de la importancia de los algoritmos en la vida cotidiana de la sociedad.

Antecedentes

En el marco de la materia de algoritmo II pretende desarrollar más a fondo los conceptos visto en las clases de la anterior asignatura, para ampliar el conocimiento sobre los procesos, las instrucciones y pasos a seguir para generar pequeños programas utilizables en la vida cotidiana y poder resolver limitantes, a la hora de desarrollar actividades.

Una de las formas para aprender es mediante juegos, esto tanto para grandes como para chicos, en la actualidad este método se implementa para aprender variedad de idiomas, o conceptos acerca de una materia en específico que podría resultar poco llamativo aprender de forma tradicional.

Un videojuego educativo es un material multimedia interactivo por medio del cual se puede aprender uno o varios temas. Una característica importante de un videojuego educativo, es que el conocimiento es adquirido de una forma implícita, es decir, los jugadores no se percatan de que al estar jugando van adquiriendo una serie de conocimientos concretos, sino que se van apropiando de éstos en el transcurso natural del videojuego.

Actualmente los videojuegos están siendo utilizados en los centros educativos, un videojuego educativo permite a los estudiantes desarrollar habilidades que de otra manera quizás no adquieran con la misma facilidad.

En este concepto surge la idea de desarrollar un juego de tipo educativo, el juego consistirá en mostrar una serie de preguntas y el usuario deberá seleccionar la respuesta correcta de esta manera el usuario avanzara de nivel. Cabe destacar que las preguntas en el juego serán acerca de conceptos de algoritmos.

Justificación.

Debido al poco conocimiento sobre los algoritmos en las personas que no están asociada con la informática o la tecnología en sí, se busca a través de un videojuego de tipo educativo generar preguntas para el usuario sobre conceptos básicos de algoritmos, esto permitirá al usuario adquirir un conocimiento básico acerca de los algoritmos, para que sirven, y tener claro que en la mayoría de procesos que desarrollamos los seres humanos estamos haciendo un algoritmo. A través de programas sencillos y menos complejos tengan la oportunidad de ampliar la comunidad interesada en la programación.

MARCO REFERENCIAL

¿Qué es un software?

Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistema de computación.

¿Qué es MVC?

El MVC o Modelo-Vista-Controlador es un patrón de arquitectura de software que, utilizando 3 componentes (Vistas, Modelos y Controladores), es una arquitectura importante puesto que se utiliza tanto en componentes gráficos básicos hasta sistemas empresariales; la mayoría de los frameworks modernos utilizan MVC (o alguna adaptación del MVC) para la arquitectura, entre ellos podemos mencionar a Ruby on Rails, Django, AngularJS y muchos otros más. En este pequeño artículo intentamos introducirte a los conceptos del MVC.

Antes que nada, me gustaría mencionar por qué se utiliza el MVC, la razón es que nos permite separar los componentes de nuestra aplicación dependiendo de la responsabilidad que tienen, esto significa que cuando hacemos un cambio en alguna parte de nuestro código, esto no afecte otra parte del mismo. Por ejemplo, si modificamos nuestra Base de Datos, sólo deberíamos modificar el modelo que es quién se encarga de los datos y el resto de la aplicación debería permanecer intacta. Esto respeta el principio de la responsabilidad única. Es decir, una parte de tu código no debe de saber qué es lo que hace toda la aplicación, sólo debe de tener una responsabilidad.

En web, el MVC funcionaría así. Cuando el usuario manda una petición al navegador, digamos quiere ver el [curso de angularjs](#), el controlador responde a la solicitud, porque él es el que controla la lógica de la app, una vez que el controlador nota que el usuario solicitó el curso de Angular, le pide al modelo la información del curso.

El modelo, que se encarga de los datos de la app, consulta la base de datos y digamos, obtiene todos los vídeos del curso de AngularJS, la información del curso y el título, el modelo responde al controlador con los datos que pidió (nota como en la imagen las flechas van en ambos sentidos, porque el controlador pide datos, y el modelo responde con los datos solicitados).

Una vez el controlador tiene los datos del curso de AngularJS, se los manda a la vista, la vista aplica los estilos, organiza la información y construye la página que vez en el navegador.

Resumamos entonces los conceptos.

Modelo

Se encarga de los datos, generalmente (pero no obligatoriamente) consultando la base de datos. Actualizaciones, consultas, búsquedas, etc. todo eso va aquí, en el modelo.

Controlador

Se encarga de... controlar, recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista.

Vistas

Son la representación visual de los datos, todo lo que tenga que ver con la interfaz gráfica va aquí. Ni el modelo ni el controlador se preocupan de cómo se verán los datos, esa responsabilidad es únicamente de la vista.

¿Qué es PHP?

PHP, sigla recursiva en inglés de PHP: Hypertext Preprocessor (procesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

Fue creado originalmente por Rasmus Lerdorf en el año 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHP, que es incompatible con la Licencia Pública General de GNU debido a las restricciones del uso del término PHP.

Ejemplo #1 Un ejemplo introductorio

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>
    <?php
```

```
        echo "¡Hola, soy un script de PHP!";
    ?>
</body>
</html>
```

En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo" (en este caso, mostrar "¡Hola, soy un script de PHP!"). El código de PHP está encerrado entre las etiquetas especiales de comienzo y final `<?php` y `?>` que permiten entrar y salir del "modo PHP".

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. No sienta miedo de leer la larga lista de características de PHP. En unas pocas horas podrá empezar a escribir sus primeros scripts.

¿Qué es Bootstrap?

Bootstrap es un framework desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web. Permite crear de forma sencilla webs de diseño adaptable, es decir, que se ajusten a cualquier dispositivo y tamaño de pantalla y siempre se vean igual de bien. Es Open Source o código abierto, por lo que lo podemos usar de forma gratuita y sin restricciones.

Ventajas de usar Bootstrap

La más genérica es que permite simplificar el proceso de maquetación, sirviéndonos de guía para aplicar las buenas prácticas y los diferentes estándares. Puedes tener una web bien organizada de forma visual rápidamente: la curva de aprendizaje hace que su manejo sea asequible y rápido si ya sabes maquetar. Permite utilizar muchos elementos web: desde iconos a desplegables, combinando HTML5, CSS y Javascript.

- Sea lo que sea que creemos, el diseño será **adaptable**, no importa el dispositivo, la escala o resolución.
- El **grid system**: maquetar por columnas nunca fue tan fácil, además, son muy configurables.
- Se integra muy bien con las principales **librerías Javascript**.

- El haber sido **creado por Twitter** nos da ciertas garantías: está muy pensado y hay mucho trabajo ya hecho. Por lo tanto, hay una **comunidad** muy activa creando, arreglando cosas, ofreciendo plugins y mucho más.
- Cuenta con **implementaciones externas** para WordPress, Drupal, etc.

MySQL es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL).

¿Que es MySQL?

se ejecuta en practicamente todas las plataformas, incluyendo Linux, Unix y Windows. A pesar de que se puede utilizar en una amplia gama de aplicaciones, MySQL se asocia más con las aplicaciones basadas en la web y la publicación en línea y es un componente importante de una pila empresarial de código abierto llamado LAMP.

MySQL, que fue concebido originalmente por la compañía sueca MySQL AB, fue adquirida por Oracle en 2008. los desarrolladores todavía pueden usar MySQL bajo la Licencia Publica General de GNU (GPL), pero las empresas deben obtener una licencia comercial de Oracle.

Los vástagos de MySQL se llaman derivados (forks). Ellos incluyen:

Drizzle - un sistema de gestión de base de datos ligero de código abierto en el desarrollo basado en MySQL 6.0.

MariaDB - un reemplazo popular "drop-in" desarrollado en la comunidad para MySQL que utiliza las API y los comandos de MySQL.

Percona Server con XtraDB - una versión mejorada de MySQL conocido por su escalabilidad horizontal.

MARCO CONCEPTUAL

Desarrollo del Juego

Para el desarrollo del juego se ha hecho uso del lenguaje de programación PHP y del framework Bootstrap. Así también se ha aplicado la programación estructurada con el modelo de desarrollo Modelo Vista Controlador.

La lógica del juego es la siguiente:

Mostrar a los usuarios una manera interactiva de aprender acerca de los conceptos básicos sobre algoritmos a través de preguntas triviales y para aquellos que ya tengan un poco de conocimiento ponerlo a prueba.

Como primer paso el usuario podrá elegir entre hacer un repaso de lo que son los algoritmos o dirigirse de una vez a la plataforma del juego.

Como primer paso el usuario debe registrarse para que pueda comenzar el juego.

Segundo paso el usuario deberá de elegir un alias para continuar.

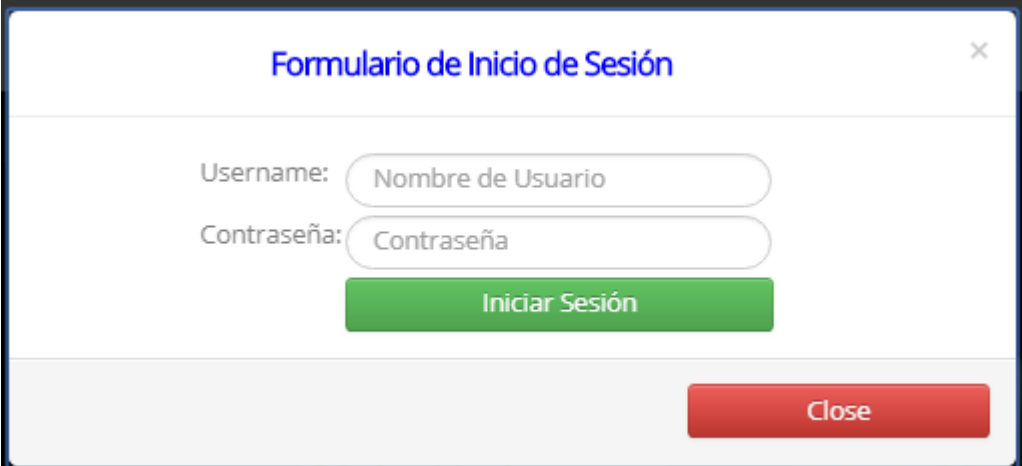
Tercer paso el usuario tendrá la opción de comenzar con el juego de preguntas reto algoritmos.

Reto algoritmo

En la pantalla de inicio se muestran algunos conceptos acerca de algoritmos y en la parte superior derecha se observan las opciones para registrarse e iniciar sesión.



En la siguiente pantalla se muestra el formulario para registrarse en el caso de que el usuario desee poner a prueba sus conocimientos. El registrar los usuarios permitirá guardar los puntajes obtenidos por cada jugador.



Formulario de Inicio de Sesión

Username:

Contraseña:

Una vez se inicie sesión se mostrara la siguiente pantalla en don se muestran cuatro opciones

1. Crear nuevo alias de jugador
2. Iniciar partida de juego
3. Ver las mejores puntuaciones obtenidas por partidas
4. Salir del juego.



A Bienvenid@ hedav

Paso 1 - Crear Nuevo Alias de jugador
Crea un nuevo Alias de Jugador, con el que se podrá Iniciar una partida de Juego.

Paso 2 - Iniciar Partida de Juego
Iniciar una nueva partida de Juego y pon a prueba tu nivel de conocimientos basicos de algoritmos.

Paso 3 - Mejores Puntuaciones
Visualiza las Mejores puntuaciones obtenidas.

Paso 4 - SALIR DEL JUEGO
Cierra la Sesión de Jugador.

Crear nuevo alias de jugador

Esta permitirá crear un alias de jugador, los alias servirán para hacer una comparación de las puntuaciones obtenidas por el jugador.

Registrar Nuevo Alias de Jugador

Nombre de Alias:

Iniciar partida de juego

Esta permitirá seleccionar un alias de jugador para iniciar una nueva partida.

Seleccionar un Alias de Jugador

Alias registrados:

Iniciar Juego

Una vez iniciada la partida se mostrara un menú en donde se mostrara el nivel a jugar, ver el progreso de la partida y finalizar partida.

Juego Reto Algoritmo

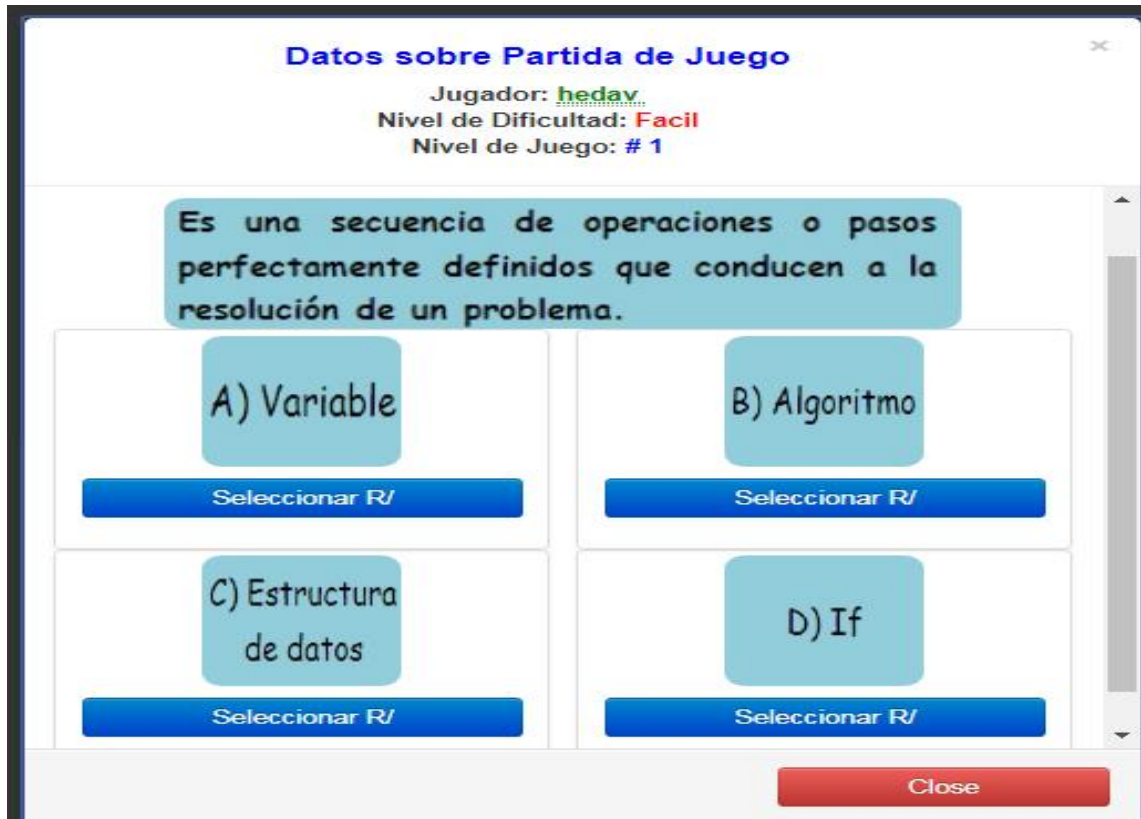
[Jugar Siguiente Nivel de Juego](#)

[Ver progreso de Partida de Juego](#)

[Finalizar Partida de Juego](#)

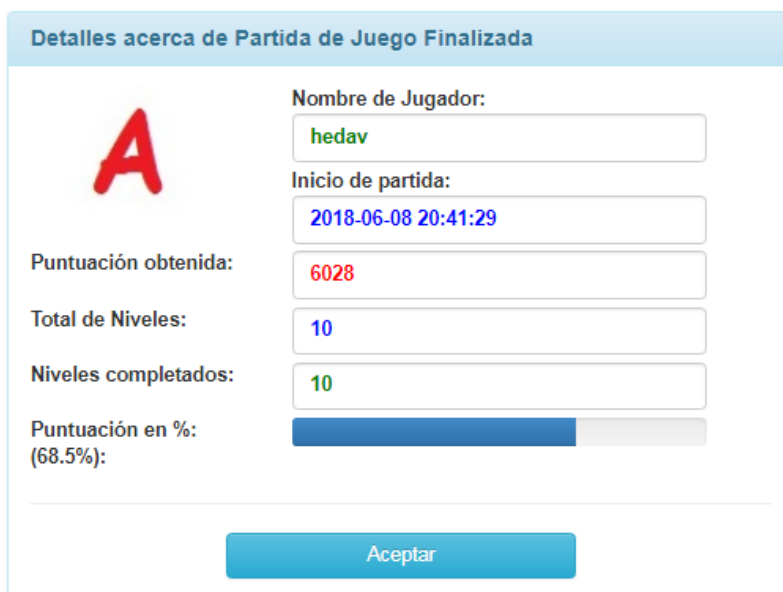
Jugar Nivel

En esta parte se muestra el nivel que se está jugando.



Puntuación final

Esta pantalla mostrara la puntuación obtenida una vez que finalice la partida.

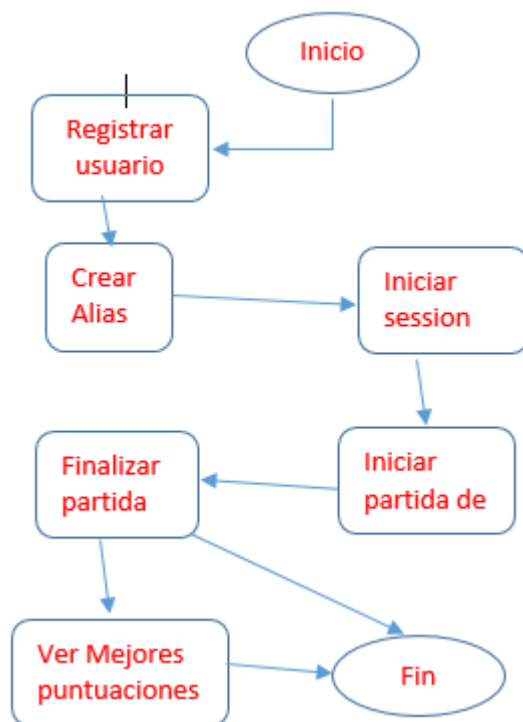


Mejores puntuaciones

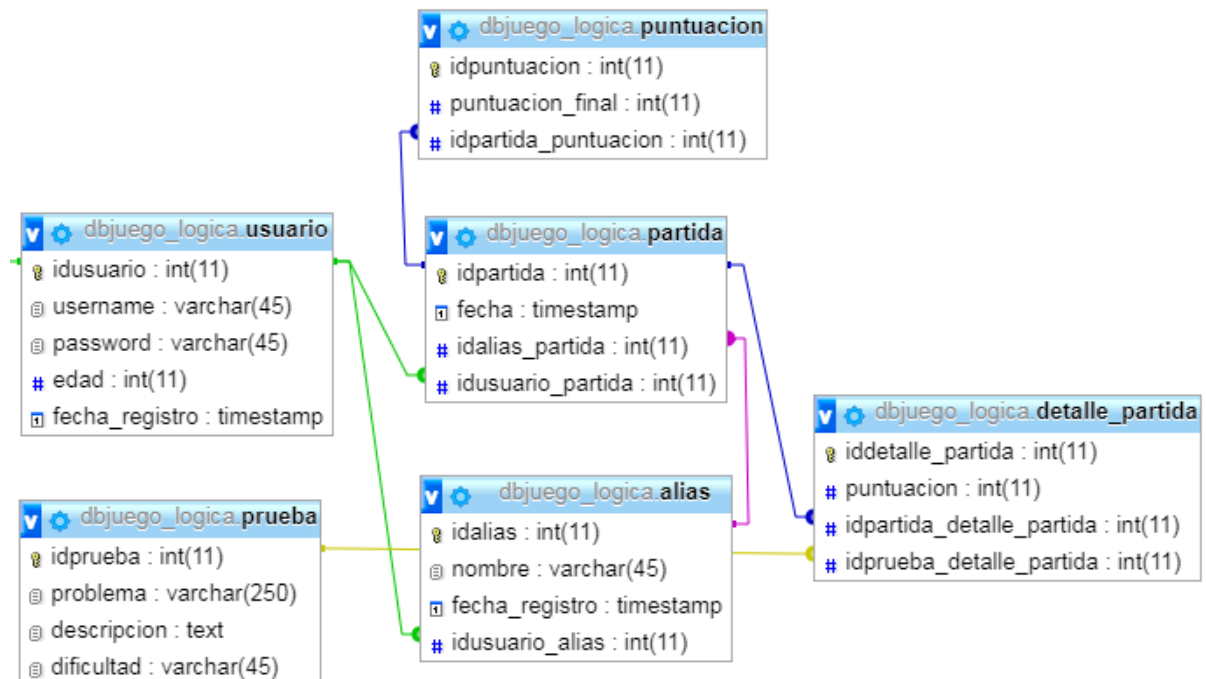
Reto Algoritmo Mejores Puntuaciones Obtenidas

Partida #	----	Jugador	Fecha de Juego	Puntuación obtenida	Detalles de Partida
28		hedav	2018-06-02 07:15:42	7336	Ver Detalles
29		henry	2018-06-02 07:24:40	6893	Ver Detalles
25		hedav	2018-06-01 21:18:05	6771	Ver Detalles
23		hedav	2018-06-01 21:06:18	6391	Ver Detalles
33		hedav	2018-06-08 20:41:29	6028	Ver Detalles

Diagrama de Flujo



Diseño de la base de datos



Código

Esta es una porción del código fuente, este código permite que el usuario desarrolle la partida de juego.

Modelo Partida

En esta parte se declaran los atributos que tendrá la partida.

```
Partida.php
1  <?php
2  class Partida {
3      //declaramos variables
4      private $idpartida;
5      private $fecha;
6      private $idalias_partida;
7      private $idusuario_partida;
8
9      // SECCION GET (PARA OBTENER LOS DATOS)
10     public function getIdpartida() {
11         return $this->idpartida;
12     }
13
14     public function getFecha() {
15         return $this->fecha;
16     }
17
18     public function getIdalias_partida() {
19         return $this->idalias_partida;
20     }
21
22     public function getIdusuario_partida() {
23         return $this->idusuario_partida;
24     }
25
26     // SECCION SET (PARA MOSTRAR LOS DATOS)
27     public function setIdpartida($idpartida) {
28         $this->idpartida = $idpartida;
29     }
30
31     public function setFecha($fecha) {
32         $this->fecha = $fecha;
33     }
34
35     public function setIdalias_partida($idalias_partida) {
36         $this->idalias_partida = $idalias_partida;
37     }
38
39     public function setIdusuario_partida($idusuario_partida) {
40         $this->idusuario_partida = $idusuario_partida;
41     }
42 }
```

Controlador Detalla de partida

Esto permitirá almacenar los datos de la partida en una base de datos.

```
Partida.php
Detalle_partidaControlador.php
1  <?php
2  class Detalle_partidaControlador extends Detalle_partida {
3      public function guardarDatos($con,$objDetalle_partida) {
4          $variableSql = "INSERT INTO detalle_partida ";
5          $variableSql .= "(iddetalle_partida,puntuacion,idpartida_detalle_par";
6          $variableSql .= "VALUES (";
7          $variableSql .= "".$objDetalle_partida[0].",";
8          $variableSql .= "".$objDetalle_partida[1].",";
9          $variableSql .= "".$objDetalle_partida[2].",";
10         $variableSql .= "".$objDetalle_partida[3].")";
11
12         return consultaA($con, $variableSql);
13     }
14
15     public function modificarDatos($con,$objDetalle_partida) {
16         $variableSql = "UPDATE detalle_partida SET ";
17         $variableSql .= "puntuacion = "".$objDetalle_partida[1]."";
18         $variableSql .= ",idpartida_detalle_partida = "".$objDetalle_partida";
19         $variableSql .= ",idprueba_detalle_partida = "".$objDetalle_partida[";
20
21         $variableSql .= " WHERE iddetalle_partida = "".$objDetalle_partida[0]";
22         return consultaA($con, $variableSql);
23     }
24 }
25 }
```

Partida Manejador

Esto permitirá definir cuáles son las respuestas correctas, en este caso se utiliza un Switch al cual se le pasara el parámetro de la respuesta seleccionada y dependiendo de la respuesta seleccionada por el usuario se le asignará una puntuación.

```
switch ($nivel) {
  case '1':
    if ($_REQUEST['opc'] != "1") { $resp = "FALSE"; } else { $resp = "TRUE"; }
    $puntuacion_obtenida = puntuaciones($grado_dif,$resp);

    break;

  case '2':
    if ($_REQUEST['opc'] != "1") { $resp = "FALSE"; } else { $resp = "TRUE"; }
    $puntuacion_obtenida = puntuaciones($grado_dif,$resp);

    break;

  case '3':
    if ($_REQUEST['opc'] != "4") { $resp = "FALSE"; } else { $resp = "TRUE"; }
    $puntuacion_obtenida = puntuaciones($grado_dif,$resp);

    break;

  case '4':
    if ($_REQUEST['opc'] != "2") { $resp = "FALSE"; } else { $resp = "TRUE"; }
    $puntuacion_obtenida = puntuaciones($grado_dif,$resp);

    break;

  case '5':
    if ($_REQUEST['opc'] != "3") { $resp = "FALSE"; } else { $resp = "TRUE"; }
    $puntuacion_obtenida = puntuaciones($grado_dif,$resp);

    break;

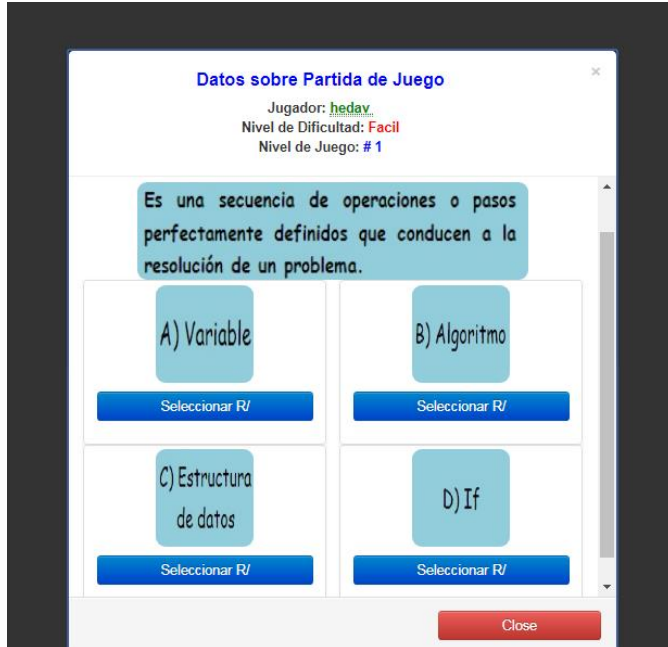
  case '6':
    if ($_REQUEST['opc'] != "3") { $resp = "FALSE"; } else { $resp = "TRUE"; }
    $puntuacion_obtenida = puntuaciones($grado_dif,$resp);

    break;

  case '7':
    if ($_REQUEST['opc'] != "1") { $resp = "FALSE"; } else { $resp = "TRUE"; }
    $puntuacion_obtenida = puntuaciones($grado_dif,$resp);
}
```

All Levels

Este formulario mostrara los niveles del juego, el formulario enviara los datos seleccionados por el usuario hacia al manejador, este evaluara si las respuestas son correctas, dependiendo del resultado se guardara una puntuación.



```
<div class="span7">
  <center>
    <h4><font color="blue">Datos sobre Partida de Juego</font></h4>
    <p></p>
    <p><b>Jugador: <font color="green">
      <abbr title="<?php print $datosAlias['nombre']. ' created: ' . $datosAlias['fecha_registro']. ' by ' . $_SESSION['username'] ?>"
      <?php print $datosAlias["nombre"]; ?>
    </abbr></font><br>
    Nivel de Dificultad: <font color="red"><?php print $datos["dificultad"]; ?></font><br>
    Nivel de Juego: <font color="blue"># <?php print $idNivel; ?></font></b></p>
  </center>
</div>
</div>
</div>
</div>
<div class="modal-body">
  <h4>Descripción breve:</h4>
  <p><?php print $datos["descripcion"]; ?></p>
  <center>
    
  </center>
  <div class="row-fluid">
    <div class="span6">
      <div class="thumbnail">
        <center>
          
          <div class="caption">
            <!--<p>Opción de Respuesta # 1</p-->
            <?php
              if ($idNivel > 0 and $idNivel < $total_de_niveles) {
                print("<p><a href='./Clases/Manejadores/manejadorPartida.php?accion=next_level&opc=1&partida=$idPartida&nivel=$idNivel&dificultad=$difi.
                  Seleccionar R/</a></p></form>");
              } elseif ($idNivel == $total_de_niveles) {
                print("<p><a href='./Clases/Manejadores/manejadorPartida.php?accion=next_level&opc=1&partida=$idPartida&nivel=$idNivel&dificultad=$difi.
                  Continuar</a></form>");
              }
            ?>
          </div>
        </center>
      </div>
    </div>
  </div>
</div>
</div>
```

CRONOGRAMA DE ACTIVIDADES

N.º	ACTIVIDADES	27/01/18	10/02/18	24/02/18	04/03/18	07/04/18	09/05/18
1	Planteamiento del problema	x					
2	Definir herramientas para el desarrollo del proyecto		X				
3	Documentar marco teórico			X			
3	Entrega del perfil del proyecto						
4	Entrega de primer avance				x		
5	Segundo avance					x	
6	Entrega final						x

CONCLUSIONES

En conclusión se puede decir que los objetivos del proyecto fueron alcanzados ya que se desarrolló una aplicación que permite conocer conceptos básicos de algoritmos. Estos forman una parte fundamental en la vida del ser humano en especial en la vida de un estudiante de computación ya que es la base para desarrollar su carrera.

Por lo tanto pretendemos generar un conocimiento básico en las personas sobre los algoritmos y su importante utilidad en el desarrollo de múltiples acciones en las cuales el ser humano es el eje principal.

Para el desarrollo de esta aplicación se implementó el modelo MVC (Modelo Vista Controlador), además del framework Bootstrap, este se utilizó para generar la parte visual de la aplicación y para el manejo de la base de datos se utilizó MySQL.

BIBLIOGRAFÍA

Febrero, 2018: <https://desarrolloweb.com/articulos/1325.php>

Febrero, 2018: <http://www.postdatastatistics.com/IntroEstadistica/TutorialesPython/Tutorial-02-py.pdf>

Febrero,
2018:http://www.academia.edu/1745600/Desarrollo_de_animaciones_en_Panda3D_con_Programaci%C3%B3n_en_Python

Marzo, 2018: <http://php.net/manual/es/intro-what-is.php>

Marzo, 2018: <https://es.wikipedia.org/wiki/PHP>

Octubre, 2018: <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>

Enero, 2015:<https://searchdatacenter.techtarget.com/es/definicion/MySQL>

