

[Dashboard](#) ▶ [My courses](#) ▶ [web2-2C-2021](#) ▶ [Exámenes](#) ▶ [\[TANDIL\] 1º Recuperatorio 15/11/2021](#)

Started on Monday, 15 November 2021, 6:00 PM

State Finished

Completed on Monday, 15 November 2021, 8:09 PM

Time taken 2 hours 8 mins

Information

MULTIPLE CHOICE

Seleccione una o varias opciones según la pregunta.

Question 1

Correct

Marked out of 100.00

MVC propone separar una aplicación en tres partes inter-relacionadas. El **modelo**, la **vista** y el **controlador**.

¿Cuáles son las funciones de cada uno?

Se encarga de la comunicación con los datos de nuestro sistema (base de datos).

modelo



Se encarga de generar la interfaz gráfica del usuario e interactuar con ellos.

vista



Coordina la comunicación entre el modelo y la vista. Acepta las peticiones del usuario y coordina el flujo de la aplicación.

controlador



Respuesta correcta

Question **2**

Correct

Marked out of 100.00

La Arquitectura Cliente-Servidor...

Select one:

- ☒ a. Es la arquitectura preponderante en los Sistemas WEB en donde un cliente realiza peticiones y un servidor se encarga de recibirlas para generar una respuesta dinámica. ✓
- ☐ b. Es el protocolo de comunicación estandar por el cual funciona la WWW.
- ☐ c. Es el modulo encargado de ejecutar los scripts del lado del servidor para generar una respuesta.
- ☐ d. TODAS LAS ANTERIORES

Respuesta correcta

Information

Complete la frase o conteste la pregunta de MANERA MUY BREVE.

No es necesario dar una respuesta muy elaborada. Es importante que la respuesta sea con sus propias palabras.

Question **3**

Complete

Marked out of 100.00

Que papel juega el protocolo HTTP en la arquitectura cliente-servidor?

Es el protocolo por el cual los clientes se comunican con el servidor, utilizando los verbos http para poder realizar operaciones, algunos de estos ejemplos son: get, post, put, delete, etc.

Comment:

Question 4

Complete

Marked out of 100.00

Cual es el papel del **servidor** en la arquitectura cliente-servidor?

Procesar las request del cliente y responder acorde. Por ejemplo, recibir una petición, realizar alguna lógica y retornar datos en formato json, un html para que se renderice en el cliente, imágenes, etc.

Comment:

Information

EJERCICIO 1 - PRÁCTICO

A raíz de la merma de shows presenciales durante la pandemia, Peter Gabriel decidió ocupar su tiempo en organizar la información de su colección de discos y hacerla disponible a sus fans a través de una plataforma web. Peter anhela que sus fans puedan encontrar inspiración en la música que él escucha. La idea es ofrecer un portal donde los usuarios puedan realizar búsquedas de álbumes y artistas y que puedan asignar valoraciones (medidas como estrellas) para los álbumes.

Peter en persona nos brinda un detalle de las tablas de la base de datos:

ARTISTA(id: int, nombre: string, premium: boolean)

ALBUM(id: int, titulo: string, productor: string, genero: string, fechaLanzamiento: string, **id_artista**: int)

VALORACION(id: int, estrellas: int, **id_album**: int, **id_user**: int)

Donde estrellas es un número entre 1 y 5.

Nota: Puede escribir el código en un IDE y copiar y pegar cada uno en el campo de texto separados por un comentario que indique el nombre del archivo.

Question 5

Complete

Marked out of 100.00

Implemente el siguiente requerimiento siguiendo el patrón MVC. No es necesario realizar las vistas ni helpers, solo controlador(es), modelo(s) y las invocaciones a la vista.

Mostrar todas las valoraciones realizadas por un usuario.

- Se deben controlar posibles errores.

```
function mostrarValoracionesRealizadasPorUsuario($params = null)
{
    if(!Auth::isLoggedIn())
        return $this->view->mostrarError('No hay un usuario logeado');

    $id_user = AuthHelper::getUserId();

    $valoraciones = $this->model->getValoracionesRealizadasPorUsuario($id_user);
    if (empty($valoraciones))
        return $this->view->mostrarError("El usuario no registra valoraciones");

    return $this->view->mostrarValoracionesRealizadasPorUsuario($valoraciones);
}

public function getValoracionesRealizadasPorUsuario($id_user)
{
    $query = $this->db->prepare('SELECT * FROM valoracion WHERE id_user = ?');
    $query->execute([$id_user]);
    return $query->fetchAll(PDO::FETCH_OBJ);
}
```

Comment:

No hay MVC solo dos metodos

Information

EJERCICIO 2 - PRÁCTICO

A raíz de la merma de shows presenciales durante la pandemia, Peter Gabriel decidió ocupar su tiempo en organizar la información de su colección de discos y hacerla disponible a sus fans a través de una plataforma web. Peter anhela que sus fans puedan encontrar inspiración en la música que él escucha. La idea es ofrecer un portal donde los usuarios puedan realizar búsquedas de álbumes y artistas y que puedan asignar valoraciones (medidas como estrellas) para los álbumes.

Peter en persona nos brinda un detalle de las tablas de la base de datos:

ARTISTA(id: int, nombre: string, premium: boolean)

ALBUM(id: int, titulo: string, productor: string, genero: string, fechaLanzamiento: string, id_artista: int)

VALORACION(id: int, estrellas: int, id_album: int, id_user: int)

Donde estrellas es un número entre 1 y 5.

Nota: Puede escribir el código en un IDE y copiar y pegar cada uno en el campo de texto separados por un comentario que indique el nombre del archivo.

Question 6

Complete

Marked out of 100.00

Implemente el siguiente requerimiento siguiendo el patrón MVC. No es necesario realizar las vistas ni helpers, solo controlador(es), modelo(s) y las invocaciones a la vista.

Buscar todos los álbumes que tengan una valoración promedio mayor o igual a cierto número de estrellas..

- Se deben controlar posibles errores.
- Se debe mostrar el artista de cada uno de los álbumes.

Nota: No es necesario hacer la acción que muestra el formulario de búsqueda.

```
public function buscarAlbunesConPromedioIgualOMayor($params = null)
{
    $promedio = $params[':PROMEDIO'];

    if ($promedio < 0 || $promedio > 5)
        return $this->view->mostrarError("El promedio debe estar entre 0 y 5");

    $albunes = $this->model->getAlbunesConValoracionPromedioMayorOIgual($promedio);

    if (empty($albunes))
        return $this->view->mostrarError("No existen albunes con promedio mayor o igual al dado.");

    $artistas = [];
    foreach ($albunes as $album) {
        $artista = $this->artistaModel->get($album->id_artista);
        array_push($artistas, $artista);
    }

    return $this->view->mostrarAlbunesConPromedioIgualOMayor($albunes, $artistas);
}

public function getAlbunesConValoracionPromedioMayorOIgual($promedio)
{
    $query = $this->db->prepare('SELECT a.id, a.titulo, a.productor, a.genero, a.fechalanzamiento, a.id_artista
    FROM album a JOIN valoracion v ON a.id = v.id_album GROUP BY a.id, a.titulo, a.productor, a.genero,
    a.fechalanzamiento, a.id_artista, v.estrellas
    HAVING AVG(v.estrellas) >= ?');
    $query->execute([$promedio]);
    return $query->fetchAll(PDO::FETCH_OBJ);
}

public function get($id)
{
    $query = $this->db->prepare('SELECT * FROM Artista WHERE id = ?');
    $query->execute([$id]);
    return $query->fetch(PDO::FETCH_OBJ);
}
```

Comment:

Information

EJERCICIO 3 - PRÁCTICO

A raíz de la merma de shows presenciales durante la pandemia, Peter Gabriel decidió ocupar su tiempo en organizar la información de su colección de discos y hacerla disponible a sus fans a través de una plataforma web. Peter anhela que sus fans puedan encontrar inspiración en la música que él escucha. La idea es ofrecer un portal donde los usuarios puedan realizar búsquedas de álbumes y artistas y que puedan asignar valoraciones (medidas como estrellas) para los álbumes.

Peter en persona nos brinda un detalle de las tablas de la base de datos:

ARTISTA(id: int, nombre: string, premium: boolean)

ALBUM(id: int, titulo: string, productor: string, genero: string, fechaLanzamiento: string, id_artista: int)

VALORACION(id: int, estrellas: int, id_album: int, id_user: int)

Donde estrellas es un número entre 1 y 5.

Nota: Puede escribir el código en un IDE y copiar y pegar cada uno en el campo de texto separados por un comentario que indique el nombre del archivo.

Question 7

Complete

Marked out of 100.00

Implemente el siguiente requerimiento siguiendo el patrón MVC. No es necesario realizar las vistas ni helpers, solo controlador(es), modelo(s) y las invocaciones a la vista.

Valorar un álbum.

- Se deben controlar posibles errores.
- Chequear que el usuario no haya valorado el álbum anteriormente.
- Si el usuario ya lo valoró, se reemplaza si la nueva es menor.

Nota: No es necesario hacer la acción que muestra el formulario de valoración.

```
function valorarAlbum($params = null)
{

    if (!Auth::isLoggedIn())
    return $this->view->mostrarError('No hay un usuario logeado');

    $id_user = AuthHelper::getUserId();

    $data = $_POST;
    $id_album = $params['ID_ALBUM'];

    if ( empty($data['estrellas']))
    return $this->view->mostrarError("No se ingresaron las estrellas.");
    $album = $this->albumModel->get($id_album);

    if (empty($album))
    return $this->view->mostrarError("No existe el album.");

    $valoracionPrevia = $this->model->getValoracionFiltradoPorUsuarioYAlbum($id_user, $id_album);
    if(empty($valoracionPrevia)){
    $this->model->create($data);
    return $this->view->mostrarNotificacion("Valoracion creada.");
    }

    if($data['estrellas'] < $valoracionPrevia->estrellas){
    $this->model->update($data);
    return $this->view->mostrarNotificacion("Valoracion actualizada.");
    }

    return $this->view->mostrarError("No se puede actualizar una valoracion por una mayor.");
}

public function get($id)
{
    $query = $this->db->prepare('SELECT * FROM Album WHERE id = ?');
    $query->execute([$id]);
    return $query->fetch(PDO::FETCH_OBJ);
}

public function getValoracionFiltradoPorUsuarioYAlbum($id_user, $id_album)
{
    $query = $this->db->prepare('SELECT * FROM valoracion WHERE id_user = ? AND id_album = ?');
    $query->execute([$id_user, $id_album]);
    return $query->fetch(PDO::FETCH_OBJ);
}
```



```
public function create($data)
{
    // con id autoincremental
    $query = $this->db->prepare('INSERT INTO valoracion (estrellas, id_album, id_user) VALUES (?, ?, ?)');
    $query->execute([$data['estrellas'], $data['id_album'], $data['id_user']]);
}

public function update($data)
{
    $query = $this->db->prepare('UPDATE valoracion SET estrellas = ? WHERE id_album = ? AND id_user = ?');
    $query->execute([$data['estrellas'], $data['id_album'], $data['id_user']]);
}
```

Comment: \$data es el arreglo \$_POST, no debería tener id_user

Information

EJERCICIO 4 – PRÁCTICO

A raíz de la merma de shows presenciales durante la pandemia, Peter Gabriel decidió ocupar su tiempo en organizar la información de su colección de discos y hacerla disponible a sus fans a través de una plataforma web. Peter anhela que sus fans puedan encontrar inspiración en la música que él escucha. La idea es ofrecer un portal donde los usuarios puedan realizar búsquedas de álbumes y artistas y que puedan asignar valoraciones (medidas como estrellas) para los álbumes.

Peter en persona nos brinda un detalle de las tablas de la base de datos:

ARTISTA(id: int, nombre: string, premium: boolean)

ALBUM(id: int, titulo: string, productor: string, genero: string, fechaLanzamiento: string, **id_artista**: int)

VALORACION(id: int, estrellas: int, **id_album**: int, **id_user**: int)

Donde estrellas es un número entre 1 y 5.

Nota: Puede escribir el código en un IDE y copiar y pegar cada uno en el campo de texto separados por un comentario que indique el nombre del archivo.

Question 8

Complete

Marked out of 100.00

Implemente el siguiente requerimiento siguiendo el patrón MVC. No es necesario realizar las vistas ni helpers, solo controlador(es), modelo(s) y las invocaciones a la vista.

Agregar un álbum al sistema.

- Se debe controlar que el usuario esté logueado al sistema.
- Se deben controlar posibles errores de carga.
- Se debe controlar que no exista un álbum con el mismo nombre.
- Si el artista tiene más de 5 álbumes se debe insertar automáticamente una valoración de 5 estrellas para ese álbum.

Nota: No es necesario hacer la acción que muestra el formulario de valoración.

```
public function agregarAlbum($params = null)
{

    if(!AuthHelper::isLogged())
    return $this->view->mostrarError("Solo un usuario logeado puede realizar esta opereación.");
    $data = $_POST;
    if (empty($data['titulo']) || empty($data['productor']) || empty($data['genero']) ||
    empty($data['fechaLanzamiento']) || empty($data['id_artista']))
    return $this->view->mostrarError("No se ingresaron todos los datos.");

    $albumConMismoNombre = $this->model->getFiltradoPorNombre($data['titulo']);

    if(!empty($albumConMismoNombre))
    return $this->view->mostrarError("Ya existe un album con ese nombre.");

    $cantidadDeAlbunesDelArtista = $this->model->getCantidadDeAlbunesDelArtista($data['id_artista']);

    $this->model->create($data);

    // se puede obtener el album por el titulo porque la consigna ya te da la pauta de que no va a haber más de un
    album con el mismo titulo
    $albumCreado = $this->model->getFiltradoPorNombre($data['titulo']);
    if($cantidadDeAlbunesDelArtista > 5)
    $this->valoracionModel->create(['estrellas' => 5,'id_album' => $albumCreado->id, 'id_user' =>
    AuthHelper::getIdUsuarioLogeado()]);

    return $this->view->mostrarNotificacion("Album creado.");
}

public function getFiltradoPorNombre($nombre)
{
    $query = $this->db->prepare('SELECT * FROM Album WHERE titulo = ?');
    $query->execute([$nombre]);
    return $query->fetch(PDO::FETCH_OBJ);
}

public function getCantidadDeAlbunesDelArtista($id_artista)
{
    $query = $this->db->prepare('SELECT COUNT(*) FROM album WHERE id_artista = ?');
```

```
$query->execute([$id_artista]);  
return $query->fetchColumn();  
}  
  
//album  
public function create($data)  
{  
    // con id autoincremental  
    $query = $this->db->prepare('INSERT INTO album (titulo, productor, genero, fechaLanzamiento, id_artista)  
VALUES (?, ?, ?, ?, ?)');  
    $query->execute([$data['titulo'], $data['productor'], $data['genero'], $data['fechaLanzamiento'],  
    $data['id_artista']]);  
}  
  
//valoracion  
public function create($data)  
{  
    // con id autoincremental  
    $query = $this->db->prepare('INSERT INTO valoracion (estrellas, id_album, id_user) VALUES (?, ?, ?)');  
    $query->execute([$data['estrellas'], $data['id_album'], $data['id_user']]);  
}
```

Comment:

Falta verificación de errores y un poco más de prolijidad

Information

EJERCICIO 5 - TEÓRICO

Question 9

Complete

Marked out of 100.00

Se quiere implementar una app de celular donde los usuarios puedan realizar sus valoraciones. Que cambios implementaría en el sistema web para que la app pueda conectarse?

- Indique que componentes del MVC agregaría o modificaría.

- Como autenticaría los usuarios?

No es necesario escribir código.

Implementaría servicios que puedan ser utilizados a través de una api rest para que se pueda interactuar con el sistema desde una aplicación mobile.

Configuraría el servidor para que pueda request desde un punto diferente como por ejemplo:

<http://artistasalbunesvaloraciones.com/api>

Agregaría las siguientes clases:

ApiController: del cual heredarían todos los controllers que se tengan que crear para manejar los servicios.

APIView: que se encargaría de manejar todas las vistas que tengan que mostrar los controladores, como por ejemplo las respuestas en formato json.

Api<servicio>Model: cualquier modelo necesario para una funcionalidad extra.

Utilizando un servicio externo como OAuth:2.0 o bien jwt, en ambos casos se requeriría que el usuario utilice un token que recibe después del login en las subsecuentes request que realice al sistema.

Comment:

Previous activity

◀ [TANDIL] Parcial 1/11/2021

Jump to...

Next activity

[Introducción a Cloud Computing ▶](#)

Stay in touch

Facultad, Pabellón Central Paraje Arroyo Seco. Campus Universitario. (B7001BBO) Tandil. Buenos Aires, Argentina

🌐 <https://exa.unicen.edu.ar/>

☎ (+54) (0249) 438-5650 Conmutador: int. 2000

✉ moodle@exa.unicen.edu.ar



📱 Get the mobile app

