

[Área personal](#) ▶ [Mis cursos](#) ▶ [web2-2C-2021](#) ▶ [Exámenes](#) ▶ [\[TANDIL\] Parcial 1/11/2021](#)

Comenzado el lunes, 1 de noviembre de 2021, 18:13

Estado Finalizado

Finalizado en lunes, 1 de noviembre de 2021, 21:12

Tiempo empleado 2 horas 58 minutos

Calificación 650,00 de 900,00 (72%)

Información

MULTIPLE CHOICE

Seleccione una o varias opciones según la pregunta.

Pregunta 1

Finalizado

Puntúa 50,00 sobre 100,00

¿Cuáles de estas son **ventajas** de tener un sistema con **enrutamiento**? (Puede elegir más de 1)

Seleccione una o más de una:

- ☒ a. Permite construir URL's semánticas para mejorar el posicionamiento web (SEO)
- ☐ b. Permite persistir los datos del usuario
- ☐ c. Permite configurar la aplicación para que acepte URL's que no mapean a archivos físicos.
- ☒ d. Reduce significativamente la cantidad de archivos de la aplicación web.
- ☒ e. Separa responsabilidades en el desarrollo del sistema web.

Comentario:

- a. Permite construir URL's semánticas para mejorar el posicionamiento web (SEO)
- c. Permite configurar la aplicación para que acepte URL's que no mapean a archivos físicos.
- d. Reduce significativamente la cantidad de archivos de la aplicación web.

Pregunta 2

Finalizado

Puntúa 100,00 sobre 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).

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

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

Comentario:

Información

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.

Pregunta 3

Finalizado

Puntúa 75,00 sobre 100,00

¿Considera que el **patrón de arquitectura MVC** es siempre la mejor opción para desarrollar un sistema web? Justifique su respuesta. Puede usar un ejemplo...

el patrón de arquitectura MVC es la mejor opción si se tiene mucho código para simplificar la vista y no perderse en tanto código al separarlo en 3 partes. por ejemplo la primera parte del tpe termina siendo mucho código entre comunicarse con la base de datos con las distintas operaciones que se pedía y llevarlo a la vista esa misma. Al aplicar MVC el código era mas fácil de ver y identificar y de seguir trabajando.

Comentario:

Falto algun ejemplo o funcionamiento de MVC

Cuando si y cuando no?

Esta mal explicado ademas el ejemplo del TPE

Pregunta **4**

Finalizado

Puntúa 80,00 sobre 100,00

Indique porque cree que PDO es más útil para conectarse a una Base de Datos.

PDO es mas útil porque no necesitas especificar a que base de datos te estas comunicando y soporta la mayoría de las bases de datos. Es mas rápido y cuando escribimos el código por ejemplo para obtener una fila termina funcionando en otras bases de datos si se llega a cambiar.

Comentario:

No habla sobre Objetos o SQL Injection

Información

EJERCICIO 1 - PRÁCTICO

La plataforma de vuelos argentina Vol.ar está desarrollando un sistema para la venta de pasajes aéreos online. La idea es ofrecer un portal donde los usuarios puedan realizar búsquedas de vuelos, consulta y compra de pasajes.

Además, quieren desarrollar una sección privada de administración, donde los agentes de viajes carguen los vuelos ofrecidos.

El departamento de IT nos brinda acceso a algunas de sus tablas de su base de datos:

AEROLINEA(id: *int*, nombre: *string*)

VUELO(id: *int*, origen: *string*, destino: *string*, fecha: *string*, estado: *string*, capacidad: *int*, internacional: *bool*, id_aerolinea: *int*)

Además, nos brinda una tabla donde se almacena la información de pasajes vendidos.

PASAJE(id: *int*, fecha_venta: *string*, clase: *int*, equipaje: *int*, id_vuelo: *int*, id_usuario: *int*)

Donde clase es un número entre el 1 y el 3, y equipaje es el peso en kilos.

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.

Pregunta **5**

Finalizado

Puntúa 100,00 sobre 100,00

Implemente la función del modelo para modificar la fecha de un vuelo.

```
<?php
//Esta seria la Carpeta Model
class FlightsModel{
    private $db;
function __construct(){
    $this->db = new PDO('mysql:host=localhost;'. 'dbname=db_vuelos_Vol_ar;charset=utf8', 'roo
}

function ModDateFlight($fecha,$id) {

    $sentencia = $this->db->prepare("UPDATE VUELO SET fecha=? WHERE id=?");
    $sentencia->execute(array($fecha,$id));

}
}
//$fecha = fecha del vuelo a modificar.
//id = id del vuelo

?>
<?php
//Esta seria la carpeta controller
require_once './Model/FLightModel.php';
require_once './View/FlightView.php';

class FlightController
{

    private $model;
    private $view;

    function __construct()
    {
        $this->model = new FLightModel();
        $this->view = new FlightView();
    }
}
```

Comentario:

```
function ModDateFlight($fecha,$id) {
    $sentencia = $this->db->prepare("UPDATE VUELO SET fecha=? WHERE id=?");
    $sentencia->execute(array($fecha,$id));
}
```

Ya solo con eso estaba bien, lo demas no lo necesitabas.

Información

EJERCICIO 2 – PRÁCTICO

La plataforma de vuelos argentina Vol.ar está desarrollando un sistema para la venta de pasajes aéreos online. La idea es ofrecer un portal donde los usuarios puedan realizar búsquedas de vuelos, consulta y compra de pasajes.

Además, quieren desarrollar una sección privada de administración, donde los agentes de viajes carguen los vuelos ofrecidos.

El departamento de IT nos brinda acceso a algunas de sus tablas de su base de datos:

AEROLINEA(id: *int*, nombre: *string*)

VUELO(id: *int*, origen: *string*, destino: *string*, fecha: *string*, estado: *string*, capacidad: *int*, internacional: *bool*, id_aerolinea: *int*)

Además, nos brinda una tabla donde se almacena la información de pasajes vendidos.

PASAJE(id: *int*, fecha_venta: *string*, clase: *int*, equipaje: *int*, id_vuelo: *int*, id_usuario: *int*)

Donde clase es un número entre el 1 y el 3, y equipaje es el peso en kilos.

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.

Pregunta 6

Finalizado

Puntúa 95,00 sobre 100,00

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

- Eliminar vuelos sin ventas

Se deberá verificar que el agente está logueado.

```
<?php
//Esta seria la Carpeta Model
class FlightsModel{
    private $db;
function __construct(){
    $this->db = new PDO('mysql:host=localhost;'. 'dbname=db_vuelos_Vol_ar;charset=utf8', 'roo
}

function DeleteFlightNotSelling() {

    $sentencia = $this->db->prepare("DELETE a.* FROM VUELO a INNER JOIN PASAJE WHERE a.id !=
    $sentencia->execute();
    // Hago que me elimine TODO los vuelos si en la tabla pasaje no hay pasaje con id_vuelo q
}

}

?>
<?php
//Esta seria la carpeta controller
require_once './Model/FLightModel.php';
require_once './View/FlightView.php';
require_once './Helpers/authHelper.php';

class FlightController
{

    private $model;
    private $view;
    private $authHelper;

    function __construct()
    {
        $this->model = new FLightModel();
        $this->view = new FlightView();
        $this->authHelper = new AuthHelper();
```

Comentario:

Correcto, faltan verificaciones de control

Información

EJERCICIO 3 – PRÁCTICO

La plataforma de vuelos argentina Vol.ar está desarrollando un sistema para la venta de pasajes aéreos online. La idea es ofrecer un portal donde los usuarios puedan realizar búsquedas de vuelos, consulta y compra de pasajes.

Además, quieren desarrollar una sección privada de administración, donde los agentes de viajes carguen los vuelos ofrecidos.

El departamento de IT nos brinda acceso a algunas de sus tablas de su base de datos:

AEROLINEA(id: *int*, nombre: *string*)

VUELO(id: *int*, origen: *string*, destino: *string*, fecha: *string*, estado: *string*, capacidad: *int*, internacional: *bool*, id_aerolinea: *int*)

Además, nos brinda una tabla donde se almacena la información de pasajes vendidos.

PASAJE(id: *int*, fecha_venta: *string*, clase: *int*, equipaje: *int*, id_vuelo: *int*, id_usuario: *int*)

Donde clase es un número entre el 1 y el 3, y equipaje es el peso en kilos.

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.

Pregunta 7

Finalizado

Puntúa 50,00 sobre 100,00

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

Comprar un pasaje para un vuelo disponible.

Un usuario logueado podrá realizar la compra de un pasaje desde un botón comprar, indicando la cantidad de personas.

Se debe verificar que el vuelo tenga disponibilidad suficiente para la solicitud.

Informar los errores que pueden aparecer.

```
<?php
//Esta seria la Carpeta Model
class FlightsModel{
    private $db;
function __construct(){
    $this->db = new PDO('mysql:host=localhost;'. 'dbname=db_vuelos_Vol_ar;charset=utf8', 'root', 'root');
}

function BuyFlight($cantidad,$fecha_venta,$equipaje,$id_vuelo,$id_usuario){
    for ($i = 1; $i <= $cantidad; $i++) {
        $sentencia = $this->db->prepare("INSERT fecha_venta=?,equipaje=?,id_vuelo=? FROM PASAJES");
        $sentencia->execute(array($fecha_venta,$equipaje,$id_vuelo,$id_usuario));
    }
}

function CheckCapacity($id,$cantidad){
    $sentencia = $this->db->prepare("SELECT capacidad from VUELO WHERE id=?");
    $sentencia->execute(array($id));
    $capacidad = $sentencia->fetch(PDO::FETCH_OBJ);
    $resultado = $capacidad - $cantidad;
    if ($resultado>=0){
        return true;
    }
    else {
        return false;
    }
}
}

?>
<?php
//Esta seria la carpeta controller
require_once './Model/FlightModel.php';
require_once './View/FlightView.php';
require_once './Helpers/authHelper.php';
```

Comentario:

Las variables salen de la nada y sin ningun chequeo

Detalle en el INSERT

Pone responsabilidades del controlador en el modelo (checkCapacity)

Faltan algunos manejos de errores (ej. vuelo inexistente)

Información

EJERCICIO 4 – PRÁCTICO

La plataforma de vuelos argentina Vol.ar está desarrollando un sistema para la venta de pasajes aéreos online. La idea es ofrecer un portal donde los usuarios puedan realizar búsquedas de vuelos, consulta y compra de pasajes.

Además, quieren desarrollar una sección privada de administración, donde los agentes de viajes carguen los vuelos ofrecidos.

El departamento de IT nos brinda acceso a algunas de sus tablas de su base de datos:

AEROLINEA(id: *int*, nombre: *string*)

VUELO(id: *int*, origen: *string*, destino: *string*, fecha: *string*, estado: *string*, capacidad: *int*, internacional: *bool*, id_aerolinea: *int*)

Además, nos brinda una tabla donde se almacena la información de pasajes vendidos.

PASAJE(id: *int*, fecha_venta: *string*, clase: *int*, equipaje: *int*, id_vuelo: *int*, id_usuario: *int*)

Donde clase es un número entre el 1 y el 3, y equipaje es el peso en kilos.

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.

Pregunta 8

Finalizado

Puntúa 0,00 sobre 100,00

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

Buscador de vuelos nacionales de una aerolínea determinada

Un usuario podrá realizar una búsqueda de vuelos desde un formulario. Se deben listar todos los vuelos que coincidan con la búsqueda. Se deben cumplir los siguientes requerimientos:

- Se debe informar si el vuelo tiene pasajeros en primera clase.

Informar los errores que pueden aparecer.

```
<?php
//Esta seria la Carpeta Model
class FlightsModel{
    private $db;
function __construct(){
    $this->db = new PDO('mysql:host=localhost;'. 'dbname=db_vuelos_Vol_ar;charset=utf8', 'root', 'root');
}

function SearchFlights($destino,$fecha,$estado,$capacidad,$internacional,$id_aerolinea){
    $sentencia = $this->db->prepare("SELECT a.* FROM VUELO a INNER JOIN PASAJE b WHERE a.fecha=? , a.estado=?, a.capacidad=?, a.internacional=?, a.id_aerolinea=? if (b.clase=1)");
    $sentencia->execute(array($destino,$fecha,$estado,$capacidad,$internacional,$id_aerolinea));
    $search = $sentencia->fetchAll(PDO::FETCH_OBJ);
    return $search;
}

}

?>

<?php
//Esta seria la carpeta controller
require_once './Model/FLightModel.php';
require_once './View/FlightView.php';

class FlightController
{
    private $model;
    private $view;

    function __construct()
    {
        $this->model = new FLightModel();
        $this->view = new FlightView();
        $this->authHelper = new AuthHelper();
    }

    function UserSearch(){
        $search = $this->model->SearchFlights();
```

Comentario:

Información

EJERCICIO 5 – PRÁCTICO

La plataforma de vuelos argentina Vol.ar está desarrollando un sistema para la venta de pasajes aéreos online. La idea es ofrecer un portal donde los usuarios puedan realizar búsquedas de vuelos, consulta y compra de pasajes.

Además, quieren desarrollar una sección privada de administración, donde los agentes de viajes carguen los vuelos ofrecidos.

El departamento de IT nos brinda acceso a algunas de sus tablas de su base de datos:

AEROLINEA(id: *int*, nombre: *string*)

VUELO(id: *int*, origen: *string*, destino: *string*, fecha: *string*, estado: *string*, capacidad: *int*, internacional: *bool*, id_aerolinea: *int*)

Además, nos brinda una tabla donde se almacena la información de pasajes vendidos.

PASAJE(id: *int*, fecha_venta: *string*, clase: *int*, equipaje: *int*, id_vuelo: *int*, id_usuario: *int*)

Donde clase es un número entre el 1 y el 3, y equipaje es el peso en kilos.

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.

Pregunta 9

Finalizado

Puntúa 100,00 sobre 100,00

Se desea implementar una API REST para exponer los datos de la plataforma a terceros.

1. Defina los endpoints necesarios para dar soporte por API REST a las tablas AEROLINEA y VUELO. No es necesario implementarlos.
2. Siguiendo el patrón MVC implemente la API REST solo para el siguiente requerimiento. No implemente los MODELOS. Puede usar la Vista de API REST brindada por la cátedra.
 - Listar todos los vuelos y listar uno solo determinado por su ID.

ACLARACIÓN: No es necesario implementar el router del sistema ni el archivo .htaccess

GET: Vol.ar/api/vuelos //obtengo todos los vuelos.

GET Vol.ar/api/vuelo/:ID //obtengo un vuelo en específico.

DELETE Vol.ar/api/vuelo/:ID //elimino un vuelo en específico.

PUT Vol.ar/api/vuelo/:ID //modifico un vuelo en específico.

POST Vol.ar/api/vuelos //Agrego un vuelo.

GET: Vol.ar/api/Aerolinea//obtengo todas las Aerolíneas.

GET Vol.ar/api/Aerolinea/:ID //obtengo una Aerolínea en específica.

DELETE Vol.ar/api/Aerolinea/:ID //elimino una Aerolínea en específica.

PUT Vol.ar/api/Aerolinea/:ID //modifico una Aerolínea en específica.

POST Vol.ar/api/Aerolinea//Agrego una Aerolínea.

//2

<?php

//apicontroller

include './Model/FlightModel.php';

include './View/ApiView.php';

class ApiFlightController {

private \$model;

private \$view;

public function __construct()

{

 \$this->model = new FlightModel();

 \$this->view = new ApiFlightView();

}

function GetFlights(){

 \$flights = \$this->model->GetFlights();

 if (!empty(\$flights)){

 return \$this->view->response(\$flights, 200);

 }

 else {

 return \$this->view->response('No hay vuelos', 404);

 }

}

function Getflight(\$params = []){

 \$id = \$params[':ID'];

 \$flight = \$this->model->GetFlight(\$id);

 if (!empty(\$flight)){

 return \$this->view->response(\$flight, 200);

 }

 else {

 return \$this->view->response("No existe el vuelo con id: \$id", 404);

```
}  
}  
?>  
<?php  
//apiView  
class APIFlightView {  
    public function response($data, $status) {  
        header("Content-Type: application/json");  
        header("HTTP/1.1 " . $status . " " . $this->_requestStatus($status));  
        echo json_encode($data);  
    }  
  
    private function _requestStatus($code) {  
        $status = array (  
            200 => "OK",  
            404 => "Not Found",  
            500 => "Internal Server Error"  
        );  
        return (isset($status[$code]))? $status[$code] : $status[500];  
    }  
}
```

Comentario:

Actividad previa

◀ [TP Especial 2021 - Parte 2](#)

Ir a...

Siguiente actividad

[Introducción a Cloud Computing](#) ▶

[Mantente en contacto](#)

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

 <https://exa.unicen.edu.ar/>

 [\(+54\) \(0249\) 438-5650](tel:+5402494385650) Conmutador: int. 2000

 moodle@exa.unicen.edu.ar



 Descargar la app para dispositivos móviles

[Facultad de Ciencias Exactas](#) – [UNICEN](#)

Contacto administradores plataforma: E-mail moodle@exa.unicen.edu.ar – Tel. [+54 0249 4385650](tel:+5402494385650) int. 2098