

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

Comenzado el lunes, 1 de noviembre de 2021, 18:10

Estado Finalizado

Finalizado en lunes, 1 de noviembre de 2021, 21:06

**Tiempo
empleado** 2 horas 55 minutos

Calificación 650,00 de 900,00 (72%)

Información

MULTIPLE CHOICE

Seleccione una o varias opciones según la pregunta.

Pregunta 1

Finalizado

Puntúa 100,00 sobre 100,00

SQL funciona en cualquier Base de Datos relacional

Seleccione una:

- ☒ Verdadero
- ☐ Falso

Comentario:

Pregunta 2

Finalizado

Puntúa 100,00 sobre 100,00

La Arquitectura Cliente-Servidor...

Seleccione una:

- ☒ 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

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 10,00 sobre 100,00

¿Considera que autenticando usuarios en una plataforma se resuelven **todos los problemas de seguridad** de un sistema? ¿Por qué?

Considero que no, ya que tambien podrian existir usuarios anónimos, que no requieran autenticarse necesitamos ver que permisos tiene., tambien para que sea seguro cada vez que se requiera hacer alguna accion (de usuario logueado) debemos chequear iniciando con session_start() ver si la variable esta seteada para que no nos entren por la url y nos realicen acciones, como borrar o ver algo que no deben ver usuarios no logueados.

Comentario: Hay otros ataques que no son solucionados con la autenticación, como la inyección sql, phishing, etc.

Pregunta 4

Sin contestar

Puntúa como 100,00

Existen sistemas WEB que puedan ser desarrollados sin utilizar **base de datos**. Describa algún ejemplo...

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 90,00 sobre 100,00

Implemente la función del modelo para modificar el estado de un vuelo.

```
class vueloModel
{
    private $db;
    public function __construct()
    {
        $this->db = new PDO('mysql:host=localhost;' . 'dbname=db_volar;charset=utf8', 'root'
    }
    function modificarEstadoDelVuelo($id)
    {
        $sentencia = $this->db->prepare("UPDATE vuelo SET estado=? WHERE id=?");
        $sentencia->execute(array($id));
    }
}
```

Comentario:

function modificarEstadoDelVuelo(\$id) { // mal, es asi:

(\$id, \$estado)

\$sentencia = \$this->db->prepare("UPDATE vuelo SET estado=? WHERE id=?"); // muy bien

\$sentencia->execute(array(\$id)); // mal, es asi:

(array(\$estado, \$id));

}

Solo con la funcion estaba bien, el resto esta de mas. Te faltó solo un parametro, asumo que no te diste cuenta porque la consulta esta bien.

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 70,00 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 que no tengan pasajeros de primera clase.

Se deberá verificar que el agente está logueado.

```
VUELOCONTROLLER:
<?php
require_once "../VueloModel/Model.php";
require_once "../PasajeModel/Model.php";
require_once "../Vueloview/View.php";
require_once "../Helpers/AuthHelper.php";

class VueloController
{

    private $model_vuelo;
    private $view;
    private $authHelper;

    public function __construct()
    {
        $this->model_vuelo = new VueloModel();
        $this->model_pasaje = new PasajeModel();
        $this->view = new VueloView();
        $this->authHelper= new AuthHelper();
    }

    function eliminarVuelos($id)
    {
        if ($this->authHelper->checkLoggedIn()) {
            $vuelos = $this->model_pasaje->buscarPasaje();
            foreach ($vuelos as $vuelo) {
                if ($vuelo->clase != 1) { //supongo que clase 1 es primera
                    $id = $vuelo->id_vuelo;
                    $eliminado= $this->model_vuelo->eliminarVuelos($id);
                    if($eliminado!=0){
                        $this->view->mostrarMensaje("eliminado con exito");
                    }else{
                        $this->view->mostrarMensaje("hubo problemas para eliminar el vuelo ");
                    }
                    $this->view->showVuelosLocation();
                }
            }
        }
    }
}
```

Comentario:

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 90,00 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.

Cargar un vuelo

El agente de viajes podrá cargar un nuevo vuelo. Se deberá verificar que el agente está logueado. Se deberá verificar que no exista un vuelo con mismo origen, destino y fecha a menos que la cantidad de pasajes vendidos sea mayor al 80%. Utilice el patrón MVC.

```
vuelo controller:

<?php
require_once "../VueloModel/Model.php";
require_once "../PasajeModel/Model.php";
require_once "../VueloView/View.php";
require_once "../Helpers/AuthHelper.php";

class VueloController
{
    private $model_pasaje;
    private $model_vuelo;
    private $view;
    private $authHelper;

    public function __construct()
    {
        $this->model_vuelo = new VueloModel();
        $this->model_pasaje = new PasajeModel();
        $this->view = new VueloView();
        $this->authHelper = new AuthHelper();
    }

    function agregarVuelo()
    {
        //que este logueado:
        if ($this->authHelper->checkLoggedIn()) {
            if (
                !empty($_POST['origen']) &&
                !empty($_POST['destino']) &&
                !empty($_POST['fecha']) &&
                !empty($_POST['estado']) &&
                !empty($_POST['capacidad']) &&
                !empty($_POST['internacional']) &&
                !empty($_POST['id_aerolinea'])
            ) {
                // Se deberá verificar que no exista un vuelo con mismo origen, desti
                $verificacion = $this->model_vuelo->verificacionVuelo($_POST['origen'], $_
                    $_POST['fecha']);

                // a menos que la cantidad de pasajes vendidos sea mayor al 80% Utiliz
```

Comentario:

El insert no debería estar dentro del foreach

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 90,00 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.

Cancelar/Devolver un pasaje para un vuelo que se había comprado

Un usuario logueado podrá realizar la cancelación de una compra de un pasaje.

Se debe verificar que el vuelo sea nacional dado que no se admiten devoluciones de vuelos internacionales.

Informar los errores que pueden aparecer.

```
charset=utf8', 'root', '');
```

```
inner join vuelo on vuelo.id=pasaje.id_vuelo WHERE vuelo.internacional=0 AND pasaje.id=?"
```

```
charset=utf8', 'root', '');
```

```
;
```

Comentario:

no chequea que el id este seteado ni maneja error de que no exista el pasaje

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

ENDPOINTS:

agregar vuelo:

verbo:POST

www.aerolinea/api/vuelo/

eliminar vuelo:

verbo:delete

www.aerolinea/api/vuelo/:ID

OBTENER VUELOS:

www.aerolinea/api/vuelo/

obtener un vuelo determinado:

www.aerolinea/api/vuelo/:iD

cambiar algun (editar) dato del vuelo:

verbo:put

www.aerolinea/api/vuelo/:ID

filtrar vuelo por destino:

www.aerolinea/api/vuelo?destino=barcelona

agregar aerolinea:

verbo:post

www.aerolinea/api/aerolinea/

eliminar aerolinea:

verbo:delete

www.aerolinea/api/aerolinea/:ID

editar aerolinea:

verbo:put

www.aerolinea/api/aerolinea/:ID

ApiVueloController:

```

<?php
require_once "../Model/VueloModel.php";
require_once "ApiController.php";

class ApiVueloController extends ApiController
{

    public function __construct()
    {
        parent::__construct();
        $this->model = new VueloModel();
        $this->view = new ApiView();
    }
    //aca van los verbos

    public function obtenerVuelos()
    {

        $vuelos = $this->model->obtenerVuelos();
        return $this->view->response($vuelos, 200);
    }

    public function obtenerVuelos($params = [])
    {
        $idVuelo = $params["id"];
        $vuelo = $this->model->obtenerVuelo($idVuelo);
        if ($vuelo) {
            $this->view->response($vuelo, 200);
        } else {
            $this->view->response("El vuelo con el id $vuelo no existe", 404);
        }
    }
}

```

APICONTROLLER:

```

<?php
require_once "../View/ApiView.php";

abstract class ApiController
{
    protected $model;
    protected $view;
    private $data;

    public function __construct()
    {
        $this->view = new ApiView();
        $this->data = file_get_contents("php://input");
    }

    function getData()
    {

        return json_decode($this->data);
    }
}

```

ApiView:

```

<?php

class ApiView{

    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