

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

**Started on** Monday, 1 November 2021, 6:13 PM

**State** Finished

**Completed on** Monday, 1 November 2021, 8:47 PM

**Time taken** 2 hours 34 mins

**Grade** 600.00 out of 900.00 (67%)

Information

### MULTIPLE CHOICE

Seleccione una o varias opciones según la pregunta.

#### Question 1

Complete

Mark 90.00 out of 100.00

¿Cuáles de estas considera que son **ventajas** de usar un Motor de Templates tipo Smarty? (puede marcar más de 1)

Select one or more:

- ☒ a. El equipo de diseñadores podría trabajar en los templates independientemente del equipo de desarrolladores.
- ☒ b. Resulta mas simple realizar cambios visuales.
- ☒ c. Se puede ocultar información a usuarios que no tengan permiso para ver ciertas secciones.
- ☐ d. Es sencillo cambiar el motor de base de datos.
- ☒ e. Genera código mas limpio y mantenible.

Comment:

Son a, b y e.

#### Question 2

Complete

Mark 0.00 out of 100.00

SQL funciona en cualquier Base de Datos relacional

Select one:

- ☐ True
- ☒ False

Comment:

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

Mark 100.00 out of 100.00

Cuál es la recomendación de seguridad a la hora de almacenar passwords de usuarios en una base de datos?  
Justifique brevemente.

Encriptarla y nunca guardar las contraseñas reales en la base de datos para que nadie pueda conocerla ni administrador de la base, solo el usuario al que le pertenece.

Comment:

Bien.

Question **4**

Complete

Mark 100.00 out of 100.00

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

Una página web estática que solo maneje html, css, o una SPA que consuma datos de una api pública, en ambos casos no requieren de una base de datos.

Comment:

Information

## 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.

Question **5**

Complete

Mark 100.00 out of 100.00

Implemente la función del modelo para modificar la clase de un pasaje.

```
function updateClase($clase, $pasajeId)
{
    $query = $this->db->prepare("UPDATE pasaje SET clase = ? WHERE id = ?;");
    $query->execute([$clase, $pasajeId]);
}
```

Comment:

Information

## 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.

## Question 6

Complete

Mark 100.00 out of 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.

```
//vuelvo a la página de inicio.
$this->view->home();
} else
    echo "No tienes permisos para hacer esta operación.";
}
}

class PasajeModel
{
    private $db;

    function __construct()
    {
        $this->db = new PDO('mysql:host=mysql-parcial2021;port=3306;dbname=db-parcial2021',
            $this->db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    }

    function getIdsVuelosConPasajerosSinClase($clase)
    {
        $query = $this->db->prepare('SELECT DISTINCT(id_vuelo) from pasaje WHERE clase <> ?;');
        $query->execute([$clase]);
        $idsVuelos = $query->fetchAll(PDO::FETCH_OBJ);
        return $idsVuelos;
    }
}

class VueloModel
{
    private $db;

    function __construct()
    {
        $this->db = new PDO('mysql:host=mysql-parcial2021;port=3306;dbname=db-parcial2021',
            $this->db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    }

    function deleteVuelo($idVuelo)
    {
        $query = $this->db->prepare("DELETE FROM vuelo WHERE id = ?");
```

Comment:

Information

### 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.

Question 7

Complete

Mark 60.00 out of 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 la fecha no haya pasado.

Informar los errores que pueden aparecer.

```
}

class VueloModel
{
    private $db;

    function __construct()
    {
        $this->db = new PDO('mysql:host=mysql-parcial2021;port=3306;dbname=db-parcial2021',
            $this->db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION));
    }

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

class PasajeModel
{
    private $db;

    function __construct()
    {
        $this->db = new PDO('mysql:host=mysql-parcial2021;port=3306;dbname=db-parcial2021',
            $this->db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION));
    }

    function getCantidadPasajesComprados($vueloId){
        $query = $this->db->prepare('SELECT count(*) FROM pasaje WHERE id_vuelo = ?;');
        $query->execute([$vueloId]);
        $cantidadPasajesComprados = $query->fetch(PDO::FETCH_OBJ);
        return $cantidadPasajesComprados;
    }

    function add($clase, $equipaje, $vueloId, $idUserario)
    {

```

Comment:

Chequea al revés if(\$vuelo['fecha'] > date("Y-m-d"))

No verifica los valores obtenidos por POST

Hace echo en lugar de llamar a la vista



No hay calls a la view desde el controller

Nulea el id

Information

#### 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.

Question 8

Complete

Mark 50.00 out of 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 cancelado con al menos 15 días de anticipación.

Informar los errores que pueden aparecer.

```
{
    $this->model = new PasajeModel();
    $this->vueloModel = new VueloModel();
    $this->view = new View();
}

function cancelarPasaje($params){
    //ya mostré en otro ejercicio como se hacía el chequeo
    if (AuthHelper::checkLoggedIn()) {
        $idPasaje = $params['ID'];
        $pasaje = $this->model->get($idPasaje);

        //supongo que dias de diferencia esta implementada en algún lado
        if(diasDiferencia($pasaje['fecha_venta'], date("Y-m-d")) < MIN_DIAS_PREVIO_CANCEL)
            echo "No se puede cancelar el vuelos pasados los dias minimos de cancelación";
        else{
            $this->model->delete($idPasaje);
            $this->view->home();
        }
    } else
        echo "No tienes permisos para realizar esta operacion.";
}

}

class PasajeModel
{
    private $db;

    function __construct()
    {
        $this->db = new PDO('mysql:host=mysql-parcial2021;port=3306;dbname=db-parcial2021',
            $this->db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    }

    function delete($idPasaje){
        $query = $this->db->prepare("DELETE FROM pasaje WHERE id = ?");
        $query->execute( [$idPasaje]);
    }
}
```

Comment:

no se comparaba con la fecha de venta sino la del vuelo

no chequea que exista pasaje

no implementa get de pasaje

## Information

## 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.

## Question 9

Complete

Mark 0.00 out of 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 todas las aerolíneas y listar una sola determinada por su ID.

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

1.

Aerolineas

url: vol.ar/api/aerolineas

verbo: get

detalle: devuelve información de todas las aerolineas

verbo: post

detalle: crea una aerolinea nueva (con los datos del body de la request)

url: vol.ar/api/aerolineas/:id

verbo: get

detalle: devuelve información de la aerolinea con id :id aerolinea

verbo: put

detalle: edita una aerolinea con id :id (usando los datos del body de la request)

verbo: delete

detalle: elimina una aerolinea con id :id

Vuelos

url: vol.ar/api/vuelos

verbo: get

detalle: devuelve información de todos los vuelos

verbo: post

detalle: crea un vuelo nuevo (con los datos del body de la request)

url: vol.ar/api/vuelos/:id

verbo: get

detalle: devuelve información de un vuelo con id :id

verbo: put

detalle: edita un vuelo con id :id (usando los datos del body de la request)

verbo: delete

detalle: elimina un vuelo con id :id

2.

class AerolineasController extends ApiController

{

public function \_\_construct()

{

parent::\_\_construct();

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

}

function get(\$params = [])

{

if (empty(\$params)) {

\$aerolineas = \$this->model->getAll();

```
return $this->view->response($aerolineas, 200);  
} else {  
    $aerolinea = $this->model->get($params[":ID"]);  
    if (!empty($aerolinea)) {  
        return $this->view->response($aerolinea, 200);  
    } else  
        return $this->view->response([], 200);  
    }  
}  
}
```

Comment:

Previous activity

◀ [TP Especial 2021 - Parte 2](#)

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](tel:+5402494385650) Conmutador: int. 2000

 [moodle@exa.unicen.edu.ar](mailto:moodle@exa.unicen.edu.ar)



 Get the mobile app

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

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