

[Área personal](#) ▶ [Cursos](#) ▶ [Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas](#) ▶ [2021](#) ▶
[web2-2C-2021](#) ▶ [Exámenes](#) ▶ [\[TANDIL\] Parcial 1/11/2021](#)

Comenzado el lunes, 1 de noviembre de 2021, 18:08

Estado Finalizado

Finalizado en lunes, 1 de noviembre de 2021, 20:30

Tiempo empleado 2 horas 22 minutos

Calificación 830,00 de 900,00 (92%)

Información

MULTIPLE CHOICE

Seleccione una o varias opciones según la pregunta.

Pregunta 1

Finalizado

Puntúa 100,00 sobre 100,00

¿Cuál es la variable que interviene en el proceso de **carga de archivos en PHP**?

Seleccione una:

- ☐ a. \$_REQUEST
- ☒ b. \$_FILES
- ☐ c. \$_POST

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 100,00 sobre 100,00

¿Considera al **lenguaje de programación PHP** la mejor opción para realizar un sistema WEB? ¿Por qué?

si bien PHP hoy en día abarca gran porcentaje de los sistemas web existentes, esto es por su antigüedad y la cantidad de sistemas ya basados en este lenguaje. Pero esto no significa que sea el mejor, habiendo actualmente otros lenguajes de programación utilizables para el backend, por lo tanto hoy en día comparte con otros competidores su posición, esto lleva a que no haya una MEJOR opción fija sino que es dependiente de las preferencias y capacidades del programador/equipo que vaya a hacer el sistema así como de la naturaleza del sistema web que se implementará. Como información extra, muchos programadores hoy en día se inclinan hacia el uso de nodeJS, ruby o python para el desarrollo de backend haciendo aún más difícil que php sea considerada la mejor opción.

Comentario:

Bien.

Pregunta **4**

Finalizado

Puntúa 70,00 sobre 100,00

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

al utilizar PDO de la forma correcta, preparando las queries y después ejecutándolas con variables bindeadas al template de query que se hizo se logra impedir la inyección de código, por lo que hace que PDO sea un método más útil por su seguridad fácilmente implementable.

Comentario:

No habla sobre interfaz o diferentes DBs

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 cambiar los kilos del equipaje.

```
function cambiarPesoEquipaje($id, $peso){  
    $sentencia = $this->db->prepare("UPDATE PASAJE SET equipaje = ? WHERE id = ?");  
    $sentencia->execute(array($peso,$id));  
}
```

Comentario:

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 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.

- Eliminar vuelos internacionales.

Se deberá verificar que el agente está logueado.

```
class VueloController{
    private $vueloModel;
    private $view;
    private $authHelper;
    private $pasajeModel;

    public function __construct(){
        $this->vueloModel = new VueloModel();
        $this->view = new View();
        $this->helper = new authHelper();
        $this->pasajeModel = new PasajeModel();
    }

    public function borrarVuelosInternacionales(){
        if ($this->authHelper->isLoggedIn()){
            $this->vueloModel->borrarVuelosInternacionales();
            $this->view->Mostrarmensaje("vuelos Borrados");}
        else{
            $this->view->Mostrarmensaje("No estas logueado");
        }
    }
}

class VueloModel{
    private $db;

    public function __construct(){
        $this->db = new PDO('mysql:host=localhost;'. 'dbname=agenciaDeVuelos;charset=utf8', '
    }

    public function borrarVuelosInternacionales(){
        $sentencia = $this->db->prepare("DELETE FROM VUELO WHERE internacional= ?");
        $sentencia->execute(array(true)); //asumo que en la tabla pasajes, los que apuntan a
        // se eliminan en cascada por la estructura de la BB
    }
}

class authHelper{
    public function isLoggedIn(){
        session_start();
        if(isset($_SESSION['logged'])){
            session_abort();
        }
    }
}
```

Comentario:

Que pasa si tenia pasajes o aerolineas asignadas

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 100,00 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.

```
class VueloController{
    private $vueloModel;
    private $view;
    private $authHelper;
    private $pasajeModel;

    public function __construct(){
        $this->vueloModel = new VueloModel();
        $this->view = new View();
        $this->helper = new authHelper();
        $this->pasajeModel = new PasajeModel();
    }

    public function agregarVuelo($origen,$destino,$fecha,$estado,$capacidad,$internacion
    if ($this->authHelper->isLogged()){
        $vuelo = $this->model->traerVuelo($origen,$destino,$fecha);
        if ($vuelo){
            if ($this->pasajeModel->getCantPasajes($vuelo->id) > ($vuelo->capacidad
                $this->vueloModel->agregarVuelo($origen,$destino,$fecha,$estado,$cap
                $this->view->Mostrarmensaje("Vuelo Agregado");
            }
            else{
                $this->view->Mostrarmensaje("Vuelo ya existente");
            }
        }
        else{
            $this->vueloModel->agregarVuelo($origen,$destino,$fecha,$estado,$capadid
            $this->view->Mostrarmensaje("Vuelo Agregado");}
        }
        else{
            $this->view->Mostrarmensaje("No estas logueado");
        }
    }
}

class PasajeModel{
```

Comentario:

Excelente.

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 100,00 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 internacionales de una fecha 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 el vuelo junto a la aerolínea.
- Se debe informar si el vuelo tiene capacidad o ya se vendieron todos sus pasajes.

Informar los errores que pueden aparecer.

```
class VueloController{
    private $vueloModel;
    private $view;
    private $authHelper;
    private $pasajeModel;

    public function __construct(){
        $this->vueloModel = new VueloModel();
        $this->view = new View();
        $this->helper = new authHelper();
        $this->pasajeModel = new PasajeModel();
    }

    public function vuelosInternacionalesXFecha($fecha){
        $vuelos = $this->vueloModel->InternacionalesXFecha($fecha);
        foreach ($vuelos as $vuelo){
            if ($this->pasajeModel->getCantPasajes($vuelo->id) == $vuelo->capacidad){
                $vuelo->estadoCapacidad = "lleno";
            }
            else{
                $vuelo->estadoCapacidad = "Hay asientos disponibles" ;
            }
        }
        $this->view->MostrarVuelos($vuelos);
        //asumo que la view mostrara de manera correcta los elementos
        //haciendo uso de todas sus propiedades.
        //Por ejemplo el mostrar el vuelo junto a la aerolinea y su estado de capacidad
        //(el primero agregado por el inner join, el segundo por la iteracion de arriba)
    }
}

class VueloModel{
    private $db;

    public function __construct(){
        $this->db = new PDO('mysql:host=localhost;'. 'dbname=agenciaDeVuelos;charset=utf8', '
    }

    public function InternacionalesXFecha($fecha){
        $sentencia = $this->db->prepare("SELECT * FROM VUELO INNER JOIN aerolinea ON vuelo.id
    )
```

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 70,00 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

XXXXXXXX/api/aerolinea (METHOD = GET) -> Traeria todas las aerolineas
 XXXXXXXX/api/aerolinea/ID (METHOD = GET) -> trae una aerolinea
 XXXXXXXX/api/aerolinea/ID/ATRIBUTO (METHOD = GET) -> trae un atributo de una aerolinea especifica
 XXXXXXXX/api/aerolinea/ID (METHOD = DELETE) -> elimina una aerolinea (si se tiene permiso en los casos de apis restringidas)

XXXXXXXX/api/aerolinea/ID (METHOD = PUT) -> modifica una aerolinea
 XXXXXXXX/api/aerolinea/ID/ATRIBUTO (METHOD = PUT) -> modifica un atributo de una aerolinea
 XXXXXXXX/api/aerolinea (METHOD = POST) -> alta de una aerolinea
 XXXXXXXX/api/vuelo (METHOD = GET) -> trae todos los vuelos
 XXXXXXXX/api/vuelo/ID (METHOD = GET) -> trae un vuelo especifico
 XXXXXXXX/api/vuelo/ID/ATRIBUTO (METHOD = GET) -> trae un atributo de un vuelo especifico
 XXXXXXXX/api/vuelo/ID (METHOD=PUT) -> modifica un vuelo
 XXXXXXXX/api/vuelo/ID/ATRIBUTO (METHOD=PUT) -> modifica un atributo de un vuelo
 XXXXXXXX/api/vuelo/ID (METHOD=DELETE) -> elimina un vuelo
 XXXXXXXX/api/vuelo (METHOD=POST) -> Alta de un vuelo

TODOS LOS LLAMADOS A APIVIEW EN EL SIGUIENTE CODIGO SON REFERENTES AL APIVIEW PROPORCIONADO POR LA CATEDRA.

YA QUE NECESITAN LOS MISMOS DATOS, LOS OTROS PUNTOS COMPARTEN MODEL (por lo que no existe apiVuelosModel, apiAerolineasModel).

```
class apiVuelosController{
    private $vueloModel;
    private $apiView;

    public function __construct(){
        $this->vueloModel = new VueloModel();
        $this->apiView = new ApiView();
    }

    public function getVuelos($params = null){
        if (isset($params[':ID'])){
            $vuelo= $this->vueloModel->getVuelo($params[':ID']);
            if ($vuelo){
                $this->apiView->response($vuelo,200);
            }
            else{
                $this->apiView->response("No se encontro el vuelo",204);
            }
        }
        else{
            $vuelos = $this->model->traerVuelos();
            if ($vuelos){
                $this->apiView->response($vuelos,200);
            }
        }
    }
}
```

```
    }  
    else{  
        $this->apiVuelo->response("No hay vuelos",204);  
    }  
}  
}
```

los endPoints a los que responderia esta funcion serian XXXXXXX/vuelo (para todos los elementos)
y XXXXXXX/vuelo/ID (para un solo elemento)

El ruteo tendria la siguiente forma con el router proporcionado por la catedra:

```
$router->addRoute('Vuelo','GET','apiVuelosController','getVuelos');  
$router->addRoute('Vuelo/:ID','GET','apiVuelosController','getVuelos');
```

Comentario:

falta la view

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