	NOMBRE:	FECHA:
	APELLIDOS:	
	MÓDULO DWES DAW2 - UD1-UD6 Examen Práctico	FEBRERO 2026

OBJETO

Se trata de programar una aplicación web en PHP para gestionar el registro de mascotas en una clínica veterinaria asociando cada mascota con su responsable(dueño). La aplicación tendrá un control de acceso (login) y se usará el patrón de diseño MVC visto en clase. Se expondrán ciertos datos de la aplicación al exterior creando servicios web mediante una interface de aplicación (API) con servicios RESTfull.

INDICACIONES GENERALES

- 1) Sube UD1-UD6_Examen-<nombre>_<apellido>.zip
- 2) La prueba se valorará de 0 a 10 puntos.
- 3) La prueba será valorada con un 0 para todos los alumnos a los que se aprecie plagio o similitud a nivel de código fuente entre sus respectivas entregas.
- 5) Se facilitará el .SQL de la BD, y el patrón MVC.
- 6) Se valora el uso adecuado de namespaces, clases, modelos, vistas y controlador (mvc).
- 6) El alumno puede usar los apuntes de clase, sus apuntes, sus prácticas y chuleta PHP


HITOS Y VALORACIONES

PORTE 1 (70%) PATRÓN MVC. Gestión de mascotas con patrón MVC (5 p)

Haz uso del patrón mvc tomando como base de tu proyecto el código fuente con el que hemos trabajado en clase (mvexamen.zip). Crea los modelos, las vistas y los controladores necesarios para la aplicación.

Para el acceso a la base de datos puedes utilizar la clase **/app/librerias/Db.php** del **mvcomposer** entregado y directamente PDO de php si te es más cómodo.

1. Listado dinámico de mascotas mascotas.
 - 1.1. Listado dinámico y estructura HTML (0,75p)
 - 1.2. Mostrar ficha de mascota. (0,75p)
2. Nuevo registro de mascota con formulario HTML y validación PHP en base a los tipos de datos en BD **(1p) (se entrega html base)**
3. Eliminar registro de mascota **(1 p) (se entrega html base)**
4. Securización de la aplicación. **(1,5 p)**
 - 4.1. Control de acceso con registros en la base de datos. (login). El controlador principal de la aplicación "Paginas" debe controlar la seguridad del sitio web haciendo uso de los métodos: **login, logout, logged**. El modelo "Veterinario" debe comprobar en la

	NOMBRE:	FECHA:
	APELLIDOS:	
	MÓDULO DWES DAW2 - UD1-UD6 Examen Práctico	FEBRERO 2026

base de datos si el usuario existe y si la clave es correcta. **(1,5p)**

Sólo el veterinario y usuario **felix** con clave **dwes2026** y email felix@veterinarios.com debe acceder a la aplicación. (tabla VETERINARIOS)


Tabla "VETERINARIOS"

ID (INT autoincremental),
NOMBRE (varchar 50),
CLAVE (varchar 50)
EMAIL (varchar 100) (Único)

- 4.2. Cada página (vista) de la aplicación debe estar securizada. Esto es, el acceso por URL debe estar prohibido y redirigir a la pantalla de login si no se ha pasado previamente por la pantalla de login y la autenticación ha sido correcta. **(0,5)**

Listado de Mascotas

Nombre	Tipo	Fecha nacimiento	Foto
Carl	gato	2013-05-07	
Torete	agaponi	2019-01-15	
Rallito	tortuga	2015-09-21	

	NOMBRE:	FECHA:
	APELLIDOS:	
	MÓDULO DWES DAW2 - UD1-UD6 Examen Práctico	FEBRERO 2026

Login

Email:

Clave:

Registrar Mascota

Nombre:

Tipo:

Fecha nacimiento:

ID Persona:


PARTE 2. (30%) SERVICIOS WEB con API RESTful . Exposición y consumo de recursos con API. (5p)

La aplicación dispondrá de una API REST desarrollada en PHP que permita gestionar mascotas. Deberá utilizar como base la arquitectura mvc vista en clase y responder en formato JSON además de utilizar autenticación HTTP Basic y respetar los métodos HTTP.

Para probar el funcionamiento de tu api usa el archivo api_test.http o el navegador.

Método	URL	Acción
GET	/api/mascotas	Listar mascotas
POST	/api/mascotas	Crear mascota
DELETE	/api/mascotas/{id}	Eliminar mascota

1. (4p) Crea una API RESTful con los siguientes endpoints:
 - a. /api/mascotas/
 - i. GET: devuelve JSON con los datos de todas las mascotas(1p)
 - ii. y POST: crea una nueva mascota en el sistema. (2p)
 - b. /api/mascota/{id}
 - i. DELETE: elimina la mascota con el {id} proporcionado. (1p)

	NOMBRE: APELLIDOS:	FECHA:
	MÓDULO DWES DAW2 - UD1-UD6 Examen Práctico	FEBRERO 2026

2. (1p) Securización API. Para poder hacer uso de los servicios web se debe enviar al servidor un usuario y contraseña válidos.

Uso adecuado de namespaces, clases, modelos, vistas y controlador (mvc).