

## Ejercicio 1) Filtrar colección de videojuegos (RA 1, 2 y 3)

En este ejercicio deberás usar el script `ejercicio1\index.php` que encontrarás dentro del proyecto base para este examen.

Al principio del script encontrarás un array bidimensional (arrays asociativos dentro de un array indexado) con los siguientes datos de videojuegos: **título**, **año de publicación** (`anio_publicacion`) y **desarrollador**

En este ejercicio deberás implementar un formulario como el siguiente:

### Elige las iniciales para filtrar:

☐A☐B☐C☐D☐E☐F☐G☐H☐I☐J☐K☐L☐M☐N☐O☐P☐Q☐R☐S☐T☐U☐V☐W☐X☐Y☐Z

| [Reiniciar](#)

El script mostrará una tabla con los videojuegos en función las letras iniciales seleccionadas (ignorando mayúsculas y minúsculas):

La primera vez que se carga la página debe mostrarse una tabla HTML con todos los videojuegos. No debe haber ninguna letra marcada por defecto.

Videojuegos por inicial

Elige las iniciales para filtrar:

☐A☐B☐C☐D☐E☐F☐G☐H☐I☐J☐K☐L☐M☐N☐O☐P☐Q☐R☐S☐T☐U☐V☐W☐X☐Y☐Z

Filtrar

Reiniciar

Título	Año publicación	Desarrollador
Animal Crossing: New Horizons	2020	Nintendo
Assassin's Creed	2007	Patrice Désilets
BioShock	2007	Ken Levine
Call of Duty	2003	Infinity Ward
Dark Souls	2011	Hidetaka Miyazaki
Elder Scrolls V: Skyrim	2011	Todd Howard
Fallout 4	2015	Todd Howard
Grand Theft Auto V	2013	Rockstar North
Halo: Combat Evolved	2001	Bungie
Mario Kart 8 Deluxe	2017	Nintendo
Mario Party Superstars	2021	Nintendo
Metroid Dread	2021	Nintendo
Minecraft	2011	Markus Persson
Pokemon Sword and Shield	2019	Nintendo
Pokémon Brilliant Diamond and Shining Pearl	2021	Nintendo
Portal	2007	Valve Corporation
Splatoon 2	2017	Nintendo
Super Mario 3D World + Bowser's Fury	2021	Nintendo
Super Mario Odyssey	2017	Nintendo
Super Smash Bros. Ultimate	2018	Nintendo
The Legend of Zelda: Breath of the Wild	2017	Nintendo
The Legend of Zelda: Link's Awakening	2019	Nintendo
The Legend of Zelda: Skyward Sword HD	2021	Nintendo

Cuando el usuario elija una o varias iniciales, y pulse en “Filtrar”, se realizará el filtrado de aquellos títulos que comiencen por la/s letra/s indicadas. Cuando el usuario pulse el enlace reiniciar, se volverán a mostrar todos los datos sin filtrar (será un enlace a la misma página).

Videojuegos por inicial

Elige las iniciales para filtrar:

☒A☐B☐C☐D☐E☐F☐G☐H☐I☐J☐K☒L☐M☐N☐O☐P☐Q☐R☐S☐T☐U☐V☐W☐X☐Y☐Z

Filtrar

Reiniciar

Título	Año publicación	Desarrollador
Animal Crossing: New Horizons	2020	Nintendo
Assassin's Creed	2007	Patrice Désilets
Mario Kart 8 Deluxe	2017	Nintendo
Mario Party Superstars	2021	Nintendo
Metroid Dread	2021	Nintendo
Minecraft	2011	Markus Persson

Para la implementación del filtrado deberás usar controles de tipo checkbox y podrán ser seleccionados varios a la vez:

```
<input type="checkbox" name="letra[]" value="A">
<input type="checkbox" name="letra[]" value="B">
<input type="checkbox" name="letra[]" value="C">
```

...

**Ayuda adicional:**

Para la generación de las letras puedes usar la función de PHP `range('A', 'Z')` que generará un array indexado con las letras de la A a la Z ([`'A'`, `'B'`, `'C'`,...]).

Para extraer la primera letra de una cadena de texto puedes usar la función `substr` o bien tratar la cadena como un array:

```
$cad='TEST';  
echo $cad[0]; //Mostrará T
```

Para convertir una cadena a mayúsculas puedes usar `strtoupper` o compararlas ignorando mayúsculas con `strcasecmp`.

## Ejercicio 2) Filtrar colección de videojuegos (RA 6)

### Creación de la base de datos:

En el proyecto base proporcionado, localiza el archivo ***ejercicio2/script.sql*** que incluye sentencias SQL para crear una base de datos (llamada *dwes\_distancia\_2223*), una tabla (llamada *videojuegos*) e insertar registros de prueba en dicha tabla. Ejecuta este *script* MySQL/MariaDB utilizando MySQL Workbench, PhpMyAdmin, NetBeans o la línea de comandos.

La tabla creada tiene la siguiente estructura:

```
CREATE TABLE videojuegos (  
  id INTEGER,  
  titulo VARCHAR(100) NOT NULL,  
  desarrollador VARCHAR(100) NOT NULL,  
  anio_publicacion INTEGER NOT NULL,  
  CONSTRAINT PK_videojuegos PRIMARY KEY (id)  
);
```

A continuación, completa los siguientes archivos para crear una aplicación web en PHP y MySQL/MariaDB:

**a) Script *ejercicio2/conexion\_bbdd.php*:** establecerá la conexión con la base de datos previamente creada, utilizando la siguiente información:

- Nombre de la base de datos: *dwes\_distancia\_2223*
- Host o servidor: *localhost*
- Usuario: *root*
- Contraseña:

**b) Script *ejercicio2/editar\_videojuego.php*:** ejecutará una consulta SQL para modificar el videojuego con identificador 3 (3, 'Red Dead Redemption 2', 'Rockstar Games', 2018) con estos nuevos datos:

- Título: *God of War*
- Desarrollador: *Santa Monica Studio*
- Año de publicación: *2018*

### Mostrar videojuego por ID

ID del videojuego:

**c) Script *ejercicio2/mostrar\_videojuego\_1.php*:** crea un formulario que recoja el identificador de un videojuego ( *id* ) y que lo envíe mediante el método POST a *mostrar\_videojuego\_2.php*. Este formulario solo recoge un único dato numérico.

### Mostrar videojuego por ID

**d) Script *ejercicio2/mostrar\_videojuego\_2.php*:** recogerá el identificador recibido de *mostrar\_videojuego\_1.php* y mostrará toda la información del videojuego con dicho identificador obteniéndola desde la base de datos. Si el

- ID: 5
- Título: *The Last of Us*
- Desarrollador: *Naughty Dog*
- Año de publicación: *2013*

identificador proporcionado no corresponda a ningún videojuego, se mostrará un mensaje indicando que el ID proporcionado no está asociado a ningún videojuego.

### Ejercicio 3) Obtener datos de videojuegos (RA 4)

En este ejercicio tienes que completar un proceso paso a paso de solicitud de datos de videojuegos **usando obligatoriamente sesiones**. Esto quiere decir que los datos se irán pidiendo progresivamente en diferentes pantallas (cada una con su formulario) y no de golpe en una sola pantalla.

Para realizar este ejercicio debes partir del script `ejercicio3/index.php` facilitado en el proyecto base proporcionado. En dicho script encontrarás los formularios ya creados y los datos necesarios. Así podrás centrarte en el uso de sesiones y validación de datos.

#### Formulario 1. Solicitar datos del desarrollador.

El primer formulario a mostrar será un formulario para seleccionar el desarrollador del videojuego. Este formulario se mostrará como primer paso cuando no hay datos almacenados todavía en la sesión:

Selecciona el desarrollador:

- Bungie
- Hidetaka Miyazaki
- Infinity Ward
- Ken Levine
- Markus Persson
- Nintendo
- Patrice Désilets
- Rockstar North
- Todd Howard
- Valve Corporation
- Otro desarrollador
- DATO ERRÓNEO ALEATORIO

La lista de desarrolladores válidos son: `$desarrolladores=["Bungie", "Hidetaka Miyazaki", "Infinity Ward", "Ken Levine", "Markus Persson", "Nintendo", "Patrice Désilets", "Rockstar North", "Todd Howard", "Valve Corporation"]`; (se incluye en el proyecto base), aunque el usuario puede también seleccionar "Otro desarrollador".

En caso de que el usuario seleccione un dato erróneo (apartado DATO ERRÓNEO ALEATORIO), se deberá mostrar un mensaje indicando que el desarrollador indicado no es válido:

**ERROR: desarrollador no esperado**

Selecciona el desarrollador:

Cuando el usuario selecciona la opción "**Otro desarrollador**" pasará al formulario 2, y cuando el usuario selecciona una de las opciones válidas del array se mostrará el formulario 3. **En ambos casos el desarrollador seleccionado de la lista se almacenará en la sesión para posterior uso.**

#### Formulario 2. Solicitar datos de otros desarrolladores

En este formulario se pregunta al usuario por el nombre del desarrollador y solo se deberá mostrar cuando el usuario ha indicado en el formulario 1 la opción "Otro desarrollador":

Indique el nombre del desarrollador:

Los datos recibidos de ese formulario serán válidos solo si el nombre del nuevo desarrollador (`desarrollador_otro`) es una cadena de al menos 2 caracteres. Si este dato es correcto, el nombre del nuevo desarrollador se almacenará en la sesión y se continuará con el formulario 3. Si no es correcto se mostrará un mensaje de error y se volverá a pedir el nombre del nuevo desarrollador.

### Formulario 3. Preguntando por el año de publicación.

En este último paso se preguntará por el año de publicación del videojuego:

Indique el año de publicación:

Los datos recibidos de ese formulario serán válidos solo si el año de publicación (`anio_publicacion`) es un número mayor de 1950. Cuando el dato es válido se almacena a la sesión y se pasa al último paso. En caso de no recibir un dato válido se muestra un mensaje de error.

### Último paso. Mostrando los datos.

*En este último paso se mostrarán los datos que se han ido almacenado progresivamente en la sesión (puedes usar `var_dump` por ejemplo) y se borrará el contenido de la sesión, lo que permitirá comenzar de nuevo el proceso.*

### Ejercicio 4 (RA 5): Búsqueda y filtrado de videojuegos en Laravel

Desarrolla una aplicación en Laravel que permita buscar y filtrar videojuegos por nombre o año de lanzamiento. Utilizarás el proyecto de Laravel ya creado en `ej4/`. Sigue los pasos descritos a continuación para completar el ejercicio.

IMPORTANTE: Todos las dependencias están ya instaladas (carpeta `vendor`).

#### Preparación:

1. Inicia XAMPP y los servicios Apache y MySQL y crea la base de datos `dawd_dwes_junio_ej4_sim`.
2. Comprueba en `ej4/.env` que la configuración de conexión a la base de datos creada antes es correcta.
3. Abre el Símbolo de Sistema: pulsa Win + R, escribe `cmd` y presiona Intro. Luego navega al directorio del proyecto `ej4\` y verifica que PHP funciona correctamente ejecutando `php --version`.
4. Aunque Composer no es necesario, puede ser que lo quieras usar. Comprueba que funciona ejecutando `composer --version` o `php composer.phar --version` en el directorio del proyecto.

#### Desarrollo del Ejercicio:

##### a) Migración y sembrado

Ya dispones de una migración `create_videojuegos_table` y del modelo `Videojuego`. Ejecuta `php artisan migrate:fresh` y luego `php artisan db:seed` para poblar la base de datos con datos iniciales. Por último, asegúrate de que la tabla `videojuegos` está creada en tu base de datos con los datos de ejemplo.

##### b) Implementación del Controlador

Crea el controlador el `VideojuegoController` y añade:

- Un método `search()` que reciba los parámetros de búsqueda desde una petición POST y utilice el modelo `Videojuego` para recuperar los datos filtrados. Este método mostrará la vista `filtrado.blade.php` con los datos de los videojuegos filtrados.
- Un método `showForm()` que cargue la vista `filtrar.blade.php` para mostrar el formulario de búsqueda.

##### c) Configuración de Rutas

En el archivo `ej4/routes/web.php`, define dos rutas:

- Una ruta GET `mostrar-busqueda/` asociada al método `showForm()` del `VideojuegoController`.
- Una ruta POST `buscar-videojuegos/` que procese los datos del formulario y esté asociada al método `search()` del `VideojuegoController`.

#### d) Diseño de las Vistas

Crea dos vistas:

- `filtrar.blade.php` en `ej4/resources/views/`, que contenga un formulario para enviar los criterios de búsqueda, nombre o año de lanzamiento, a la ruta POST `buscar-videojuegos/`.
- `filtrado.blade.php` en `ej4/resources/views/`, que muestre los resultados de la búsqueda basándose en los datos recuperados por el método `search()`.

#### Pruebas y verificación:

1. Abre la terminal, entra en `ej4/` e inicia el servidor con `php artisan serve`.
2. Accede con tu navegador a `http://localhost:8000/mostrar-busqueda/`, y prueba las funcionalidades de búsqueda y filtrado para asegurarte de que los resultados se muestran correctamente según los criterios especificados.

### **Ejercicio 5 (RA 7, 8 y 9): Búsqueda y filtrado de videojuegos en Laravel**

---

En este ejercicio deberás completar un servicio web sin usar ningún framework y crear un pequeño cliente de dicho servicio web usando GuzzleHttp. La información que maneja tanto el servicio web como el cliente son datos de videojuegos: nombre (nombre) y año del videojuego (year).

El servicio web está a medio implementar, como puedes ver en el archivo `serve.php`, por lo que solo tienes que realizar una parte.

En `serve.php` cuando se recibe una petición POST (`$_SERVER['REQUEST_METHOD'] === 'POST'`) se espera que los datos recibidos sean tipo JSON. Como puedes ver en el código, los datos en crudo (sin procesar) se cargarán en la variable `$datosJSONrecibidos`. Deberás completar el código para esta casuística en `serve.php` (petición POST) para que:

- Se decodifiquen los datos JSON recibidos (almacenados como string en `$datosJSONrecibidos`).
- Si se ha podido decodificar el JSON de forma correcta y contienen los datos “nombre” y “year” (año del videojuego), se guardarán en el archivo serializado usando la función facilitada `guardardatos` sin machacar los datos ya existentes.

Una vez implementado ese fragmento de código en `serve.php` deberás completar el código de `consumer.php`. Como podrás observar en `consumer.php` parte de la funcionalidad ya está implementada. Tu objetivo es implementar que cuando se reciban los datos del formulario en `consumer.php` se realice una petición POST al servicio web (`serve.php`) con los datos del juego en formato JSON con el objetivo de insertarlo dentro de listado de videojuegos.