

# Aplicaciones Web - Curso 2022/2023

Convocatoria ordinaria (16/01/2023) - Duración: **3 horas**

Puntuación máxima del examen: **10 puntos**

Puntuación mínima en el examen para poder aprobar la asignatura: **4 puntos**

## Modelo A

### INSTRUCCIONES

---

1. Descarga el fichero "recursos\_examen.zip" y cópialo en un directorio donde tengas privilegios de escritura. En el fichero encontrarás la carpeta "**aw\_22\_23**" que contiene todas las carpetas necesarias para la comenzar a realizar el examen, la carpeta **node-modules** y los ficheros **package.json** y **package-lock.json**. También verás una carpeta llamada "**recursos**" con un conjunto de archivos que tendrás que ir colocando en la carpeta correcta del proyecto. Cuando ya hayas colocado estos archivos puedes eliminar esta carpeta de tu carpeta de examen.
2. Abre VS Code y comprueba que la carpeta donde realizarás el examen está correctamente formada.
3. Los ejercicios que se vaya a entregar debe funcionar, cumpliendo los objetivos requeridos en menor o mayor medida, pero no se admitirán errores de recursos no encontrados (páginas, imágenes, etc..). Si un ejercicio entregado lanza cualquier error de este tipo la calificación será de 0 en el ejercicio correspondiente.
4. El diseño es importante, intenta que los resultados se muestren de forma clara y ordenada.
5. En la corrección se valorará el funcionamiento, la claridad del código y el diseño de las soluciones.
6. Como ayuda al código necesario en cada ejercicio se aporta las siguientes líneas:

```
"use strict";

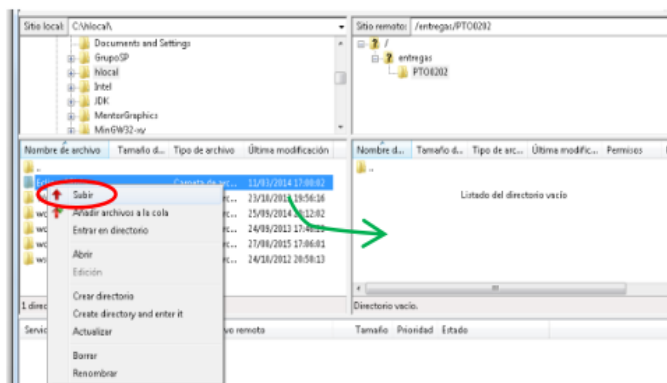
const path = require("path");
const express = require("express");
const app = express();
app.set("view engine", "ejs");
app.set("views", path.join(__dirname, "views"));
app.use(express.json());
```

## INSTRUCCIONES DE ENTREGA

---

Para entregar la solución del examen, crea un fichero **NombreApellidos.zip** (usar sólo **ZIP**, no **RAR**, ni **7z**). En él debes incluir todo el proyecto quitando una vez limpiado de archivos intermedios (es decir, sin el directorio **bin**, que contiene los **.class**). Haz doble clic en el icono del escritorio denominado: “**EXAMENES en**

**LABs entregas**”, y dentro de la ventana que aparece, haz doble clic en “**ALUMNOS entrega de prácticas y exámenes**”. Se abre otra ventana en la que debes seleccionar el archivo **zip** en el panel inferior izquierdo y arrastrarlo al panel inferior derecho (o utiliza el botón derecho del ratón y la opción **Subir**), como se indica en la figura. Antes de abandonar el laboratorio debes pasar por el puesto del profesor para asegurarte de que lo que se ve en el puesto del profesor es lo que has entregado (**comprobando el tamaño del archivo**) y firmar en la hoja de entregas (usar tu bolígrafo para evitar el contacto).



## Pregunta 1 [3,5 puntos]

---

Crear un archivo llamado `ejerc1.js` que contendrá la solución a este problema.

En la Intranet de la empresa se ha decidido establecer una nueva funcionalidad de cambio de contraseña en el formulario de acceso al sitio web. Se trataría de establecer un nuevo enlace dentro de la caja de identificación con el texto “Cambio de contraseña”.



Formulario de acceso (Login) con el título "Acceso".

Campos:

- Login:
- Password:

Enlaces y botones:

- [Cambiar contraseña](#)
- 

Para mantener el nivel de seguridad que exige la empresa, el usuario debe primero introducir su login y password actual de forma que antes de abrir la nueva ventana de cambio de contraseña se compruebe que el usuario es válido y puede proceder a cambiar su contraseña, en caso contrario se mostraría una línea de color rojo indicándole que “El usuario o contraseña son incorrectos”. En el caso en que los datos sean correctos se abrirá una nueva página con la dirección **localhost:3000/cambio**.

La nueva página solicita al usuario que introduzca la nueva clave y por seguridad la repita. Las dos introducciones de clave deben ser iguales, la longitud se estima que debería ser mayor de 5 caracteres y debe contener 3 mayúsculas y 2 dígitos.

Para este ejercicio simularemos la base de datos con una variable del tipo:

```
var usuarios = [  
  {nombre: "Aitor", apellidos: "Tilla Patata", user:"aitor01", pass:  
    "ATP01"},  
  {nombre: "Carmelo", apellidos: "Cotón Partido", user:"carmelo02", pass:  
    "CCP02"},  
  {nombre: "Mirella", apellidos: "Baila Sola", user:"mirella03", pass:  
    "MBS03"},  
  {nombre: "Felipe", apellidos: "Lotás Blandas", user:"felipe04", pass:  
    "FLB04"}  
]
```

Al menos una gestión de errores 404 y 500 es necesario en este ejercicio.

## Pregunta 2 [3,5 puntos]

---

Crear un archivo llamado `ejerc2.js` que contendrá la solución a este problema.

En plena campaña de Navidad una cadena de tiendas encarga un aplicación para el control de su clientela, de manera que una aplicación web recoja datos de los clientes que han realizado una compra en sus establecimientos. Para ello cuentan con un punto de información en el mostrador que es simplemente un ordenador con una página web cargada. La dirección de entrada a esta aplicación es **localhost:3000/**

Cada establecimiento tendrá su propia aplicación corriendo y al final del día cada uno comunicará a la central el total de datos recogidos.

A la dirección de marketing de la empresa le interesa conocer tres datos fundamentales:

- El rango de edad: menor que 20, de 20 a 30, de 30 a 40 y mayor de 40
- El canal por el que les ha conocido: Buscador de internet, Redes sociales, campaña de prensa y tv, amigos
- El lugar de residencia del cliente: Madrid, Comunidad de Madrid, resto del territorio o fuera de España.

Cada vez que un cliente envíe los datos la aplicación mostrará durante una página de agradecimiento, donde figure un mensaje de gracias que será una página accesible en la dirección **localhost:3000/gracias** y en 5 segundos volverá a cargar la página de la encuesta para que otro cliente pueda contestar. Para ello se puede hacer uso del código de redireccionamiento que se indica a continuación:

```
<body onLoad="redireccionar()">
...
<script>
    function redireccionar() {
        setTimeout("location.href='http://localhost:3000/'", 5000);
    }
</script>
```

Los datos no se almacenarán en ningún sitio, simplemente al finalizar la jornada, el responsable de la tienda accederá a los resultados mediante la dirección **localhost:3000/resultados**, tomará nota de los resultados y los comunicará por otro medio no especificado a las oficinas centrales.

### Pregunta 3 [3 puntos]

---

Crear un archivo llamado `ejerc3.js` que contendrá la solución a este problema.

- Levantar un servidor de Express en el puerto 3000
- Al llamar a la dirección **localhost:3000/ofertas** debe mostrar lo más atractivamente posible el siguiente json:

```
{
  descripcion: 'Productos',
  items: [
    {nombre: 'taza de Star Wars', precio: 300, imagen: 'taza-stars-war.jpg' },
    {nombre: 'FIFA 22 PS4', precio: 100, imagen: 'fifa22_PS4.jpg'},
    {nombre: 'Camiseta Marbel', precio: 100, imagen: 'camiseta-marvel.jpg' },
    {nombre: 'Figura Funco Stranger', precio: 200, imagen: 'funco-lucas-stranger.jpg'},
    {nombre: 'Botella Marvel', precio: 120, imagen: 'botella-marvel.jpg' },
    {nombre: 'Llavero Star-Wars', precio: 120, imagen: 'llavero-star-wars.jpg'},
  ]
}
```

Para ello se dispone de las imágenes en la carpeta **recursos** las cuales habrá que situar correctamente en la estructura de carpetas del proyecto.