	Curso / Kurtsoa	Fecha / Data	Nivel / Maila	Eval. /Ebal.
	2012-23	09/02/2023	2º	2ª
Módulo / Modulua	Kodea / Código	U.Didak /Unid didác	tipo	Calificación/Kalifikazioa
DWC	2DW3	1, 2, 3, 4, 5, 6, Vue.js	P	
Nombre Alumno/a / Ikaslearen izena				

EXAMEN MINIMOS

- Este examen es de tipo procedimental.
- Tiene un valor total de 12.
- Si la propuesta del examen se resuelve de forma correcta se le aplica el valor especificado en cada apartado.
- Crear cada ejercicio en un directorio diferente con el nombre Ejercicio1, Ejercicio2... Dentro de cada directorio meter todos los archivos que necesitéis para hacer el ejercicio.
- Cuando terminéis, subir los archivos comprimidos en el drive que tenéis a vuestro nombre (cada ejercicio en su carpeta), con el nombre EVA2_Apellidos_Nombre.

Ejercicio 1. Javascript: (2 ptos)

Hemos generado un código utilizando Inteligencia Artificial para un enunciado que dice algo así:

Quiero una página web en HTML5 con un formulario que tenga lo siguiente:

- *Un campo input con id y name "matricula" con una expresión regular que obligue a introducir cuatro números, un guión y tres letras que no pueden ser vocales.*
- *Un campo select con un id y name "tipo" que tenga tres opciones: motocicleta, automovil o camion.*

Además, habrá seis imágenes ocultas de 100px por 100px con identificadores: menorEdad, mayorEdad, particular, profesional, rigido o articulado.

Si el campo select cambia y se elige la opción "motocicleta", se mostrarán las imágenes "menorEdad" y "mayorEdad", si se elige la opción "automovil" se mostrarán las imágenes "particular" y "profesional", y si se elige la opción "camion" se mostrarán las imágenes "rigido" y "articulado" utilizando Javascript.

Además, si se hace clic sobre cualquiera de las imágenes se activará o desactivará una clase "bordeRojo" que estará establecida en el CSS del head de la página y que permitirá establecer un borde rojo de 3 píxeles alrededor de la imagen pulsada o se desactivará en caso de que se vuelva a pulsar.

Al final del formulario habrá un botón que permitirá mostrar, debajo, los siguientes datos:

- *Matrícula: mostrará el contenido del campo input.*
- *Automóvil: mostrará la opción elegida en el campo desplegable.*
- *Tipo de automóvil: mostrar el atributo alt de las imágenes marcadas en rojo.*

Sin embargo, el resultado obtenido no nos gusta demasiado y queremos hacer algunos retoques:

1. Es posible marcar dos o más imágenes con el borde rojo (incluso si están ocultas), así que en el apartado "Tipo de automóvil" se muestran más de una imagen. Solamente debería haber una imagen marcada con borde rojo de cada vez.

- En la sección "Ocultar todas las imágenes al inicio" hay seis líneas que tienen el método `"style.display = 'none';"`. Necesitamos optimizar esta parte para que el método se aplique a todas las imágenes, independientemente del número de imágenes que haya. Puedes tocar todo el código que quieras, incluido el que ya está escrito.
- Lo mismo ocurre en la parte de añadir el borde rojo a las imágenes. Debería aplicarse a todas las imágenes de una manera más eficiente que haciendo 6 funciones.
- En la última función (cuando hacemos click en el botón), queremos que en caso de que no haya ninguna imagen seleccionada, que se muestre un mensaje "No se ha seleccionado ninguna imagen".

Total ejercicio - 1	2
Solamente se marca una imagen con borde rojo cada vez.	1
Se ha optimizado la parte de Ocultar todas las imágenes al inicio.	0,5
Se ha optimizado la parte de añadir el borde rojo a las imágenes.	0,3
Se ha añadido el mensaje final cuando no se ha añadido ninguna imagen.	0,2

Ejercicio 2. Vue.js: (5 ptos)



En este ejercicio utilizaremos la librería Axios que está en la carpeta `./Datos/Axios/axios.min.js` y el fichero json que está en `./Datos/pokemons.json`

2.1.- Al comienzo debes mostrar en una tabla el contenido del array con los encabezados Nombre, Tipo y Habilidad. Los datos se recogen del archivo `./Datos/pokemons.json`

2.2.- Añadiremos también una columna en cada fila, con un botón. Este botón nos eliminará el pokémon del array.

CRUD DE POKEMONS			
Nombre	Tipo	Habilidad	
bulbasaur	plant	Razor Leaf	
ivysaur	plant	Seed Bomb	
charmeleon	fire	Flamethrower	
charizard	fire	Inferno	
squirtle	water	Water Gun	
metapod	bug	Harden	
butterfree	bug	Stun Spore	
pidgeotto	flying	Wing Attack	
pidgeot	flying	Air Slash	

2.3.- Al final de la tabla pondremos los campos necesarios para añadir un pokémon al array (no se modifica el fichero .json).

arbok	poison	Toxic	
pikachu	electric	Thunder Shock	

Añadir Pokemon

Nombre: Tipo: Habilidad: 

2.4.- Para terminar, podremos filtrar los pokémons a partir de su tipo. (Podemos usar campo select, para seleccionar el tipo) :

Filtrar Pokemons

plant ▾		
Nombre	Tipo	Habilidad
bulbasaur	plant	Razor Leaf
ivysaur	plant	Seed Bomb

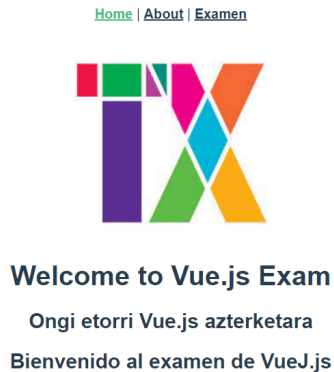
Total ejercicio - 2	5
Crear tabla inicial con los datos del json	1,5
Crear botón de eliminar	0,25
Al pulsar eliminar => se borra el registro	1
Crear formulario añadir pokemon	0,25
Botón añadir pokemon => se añade un nuevo Pokemon al listado	1
Campo select para filtrar por tipo de pokemon => no se repiten los tipos en select	0,5
Filtra los pokemon de ese tipo	0,5

Ejercicio 3. Vue.js: (5 ptos)

En este ejercicio vamos a tomar como base el proyecto ya creado y que está en la carpeta.

Como veis en este proyecto tenemos 2 rutas que apuntan a las vistas **HomeView.vue**(en la que se carga el componente HelloWorld.vue), y la vista **AboutView.vue**. Tendréis que hacer lo siguiente

1. Modificar Home para que se vea lo siguiente. El logo *TX.jpg*, lo tenéis en la carpeta de enunciado



2. Crear una nueva vista **ExamenView.vue** y en ella hacer lo siguiente: Hay que crear un CRUD local con los datos del alumnado.

[Home](#) | [About](#) | [Examen](#)

Aquí montamos el examen

Examen mínimos 2DW3

Nombre:
Email:
Password:

- a. Al inicio aparece un formulario para introducir el nombre, correo electrónico y password.
- b. **Nombre:** acepta cualquier tipo de dato
- c. **Email:** se valida que sea un correo electrónico válido. Al hacer blur se comprueba, y si es correcto pasamos al campo siguiente; si no es correcto, vuelve a obtener el foco y aparece un mensaje indicando **correo incorrecto**

[Home](#) | [About](#) | [Examen](#)

Aquí montamos el examen

Examen mínimos 2DW3

Nombre:
Email: **correo incorrecto**
Password:

- d. **Password:** se verifica que la contraseña tenga mayúsculas, minúsculas y números, con una longitud de entre 8 y 16 caracteres. Según vayamos escribiendo, es decir, ante el evento keyup (o keypress o keydown), irá detectando que la contraseña es correcta o no. Si es incorrecta me saldrá el mensaje **Password incorrecto**. Cuando el password sea correcto aparecerá el botón de **Nuevo alumno/a**

[Home](#) | [About](#) | [Examen](#)

Aquí montamos el examen

Examen mínimos 2DW3

Nombre:

Email:

Password:

- e. Al pulsar el botón se añadirán los datos en el array y se renderizarán en una tabla que mostraremos en la parte inferior de la pantalla.

[Home](#) | [About](#) | [Examen](#)

Aquí montamos el examen

Examen mínimos 2DW3

Nombre:

Email:

Password:

Nombre	email	Password		
Jose	jminsausti@fpburdinaga.com	2dw3-Jose	<input type="button" value="Eliminar"/>	<input type="button" value="Activar Edición"/>
Laura	laura.f@fpburdinaga.com	2dw3-Laura	<input type="button" value="Eliminar"/>	<input type="button" value="Activar Edición"/>
Txomin	asdf@gmail.com	Asdf1234	<input type="button" value="Eliminar"/>	<input type="button" value="Activar Edición"/>

- f. Junto a los datos de cada alumno/a, se mostrarán dos botones: **Eliminar** y **Activar edición**. Si pulsamos **Eliminar**, se eliminará la información de ese/a alumno/a.

[Home](#) | [About](#) | [Examen](#)

Aquí montamos el examen

Examen mínimos 2DW3

Nombre:

Email:

Password:

Nombre	email	Password		
Laura	laura.f@fpburdinaga.com	2dw3-Laura	<input type="button" value="Eliminar"/>	<input type="button" value="Activar Edición"/>
Txomin	asdf@gmail.com	Asdf1234	<input type="button" value="Eliminar"/>	<input type="button" value="Activar Edición"/>

- g. Si pulsamos **Activar edición**, los campos de las tablas se volverán editables, y en el botón aparecerá la leyenda **Desactivar edición**. Al modificar la información de cualquiera de los campos y pulsar Desactivar edición, volveremos a la situación anterior, pero con esa información modificada.

Aquí montamos el examen

Examen mínimos 2DW3

Nombre:
Email:
Password:

Nombre	email	Password		
Laura	<input type="text" value="laura.f@fptxurdinaga.com"/>	<input type="text" value="2dw3-Laura"/>	<input type="button" value="Eliminar"/>	<input type="button" value="Desactivar Edición"/>
Txomin	<input type="text" value="asdf@gmail.com"/>	<input type="text" value="Asdf1234"/>	<input type="button" value="Eliminar"/>	<input type="button" value="Desactivar Edición"/>

Aquí montamos el examen

Examen mínimos 2DW3

Nombre:
Email:
Password:

Nombre	email	Password		
Laura	<input type="text" value="laura.f@fptxurdinaga.com"/>	<input type="text" value="2dw3-Laura"/>	<input type="button" value="Eliminar"/>	<input type="button" value="Activar Edición"/>
Txomin Cambiado	<input type="text" value="asdf@gmail.com"/>	<input type="text" value="Asdf1234"/>	<input type="button" value="Eliminar"/>	<input type="button" value="Activar Edición"/>

Total ejercicio - 2	5
Modificar Home para mostrar logo TX y mensajes	0,25
Crear ExamenView.vue y que sea accesible desde las rutas	0,25
Mostrar formulario	0,25
Email validado	0,25
Blur en email incorrecto => mensaje de correo incorrecto	0,75
Email incorrecto => focus de nuevo en el campo	0,25
Password keypress => mensaje Password incorrecta	0,75
Password correcta => se muestra botón de Nuevo alumno	0,5
Pulsar nuevo alumno => se muestran en la tabla	0,5
Pulsar Eliminar => borra la fila	0,5
Pulsar Activar edición => campos editables	0,5
Pulsar Activar edición => cambio en nombre del botón	0,25

