|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mostrando Marca_Txurdinaga_Color_H_SUP_01_01.jpg | Curso / *Kurtsoa* | Fecha / *Data* | Nivel / *Maila* | Eval. /*Ebal.* |
| 2024-25 | 21/05/2025 | 2º | 2ª evaluación |
| 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**

* Este examen es de tipo procedimental.
* Tiene un valor total de 10 puntos.
* Si la propuesta del examen se resuelve de forma correcta se le aplica el valor especificado en cada apartado.
* Descargar el scaffolding desde Github Classroom. En tu equipo, haz los commits necesarios. Cuando quieras entregar sube el repositorio a Github.

**Ejercicio 1. Javascript(2 ptos)**

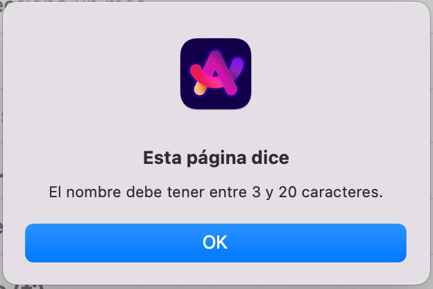
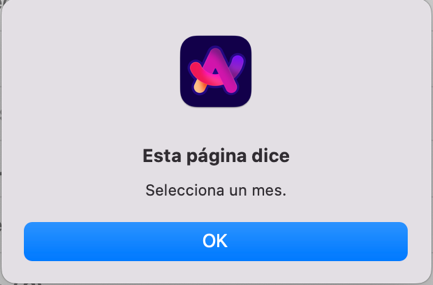
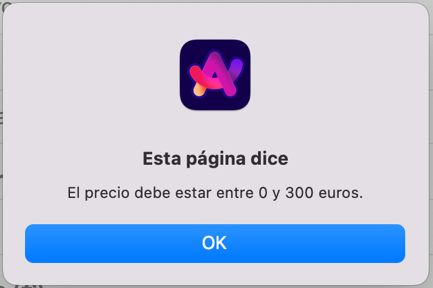
Dispones de un formulario en HTML para introducir festivales en una página web de festivales del mundo.

1. **Validaciones (JavaScript):**

Implementa una función que valide los campos antes de procesar el formulario:

* El nombre del festival y el país deben tener entre **3 y 20 caracteres**. Llamarás a una función por cada uno de estos dos campos en la que le pases un texto, un número mínimo y uno máximo y devuelva true o false si es correcto o no.
* El mes y el género musical deben estar seleccionados (no pueden estar vacíos).
* El precio debe estar comprendido entre **0 y 300**.

En caso de que la validación sea correcta se crea un objeto Festival. En caso de que sea incorrecta, mostrará un mensaje con el primer error que tenga el formulario.

1. **Creación de la clase Festival:**

* Crea una clase Festival con los atributos: nombre, mes, país, género, y precio.
* Añade un método mostrar() que devuelva una cadena HTML con la información del festival.

1. **Visualización de los datos**

* Si los datos son válidos, crea una tarjeta con la información del festival debajo del formulario, dentro de la etiqueta cuyo id es “datos” (<div id=”datos”></div>) para lo que se llama al método mostrar de la clase Festival.
* Cada nuevo festival se añade debajo del anterior, sin borrar los existentes.

|  |  |
| --- | --- |
| **Total ejercicio - 1** | **2** |
| * 1. Se ha creado la función de validación del texto tal y como se solicita y funciona correctamente. | 0,25 |
| * 1. Se han validado los campos de texto correctamente con JS. | 0,25 |
| * 1. Se han validado los campos desplegables (no están vacíos) correctamente con JS. | 0,10 |
| * 1. Se ha validado el campo de precio correctamente con JS. | 0,10 |
| * 1. En caso de que el formulario no se valide correctamente, se muestra el primer error que encuentra. | 0,10 |
| * 1. En caso de que el formulario se valide correctamente, se crea un objeto de tipo Festival. | 0,25 |
| * 1. El constructor de la clase Festival.js es correcto. | 0,25 |
| * 1. El método mostrar es correcto. | 0,25 |
| * 1. En caso de que el formulario se valide correctamente, se añade la información del festival en la etiqueta “datos”. | 0,25 |
| * 1. Si se valida más de un formulario, la información se añade al final de la etiqueta “datos” | 0,20 |

**Ejercicio 2. Vue.js: (8 ptos)**

La página web Festival Hub recoge algunos de los festivales más importantes del panorama musical. A través de ella podremos visualizar todos los festivales, realizar búsquedas, filtrar por género, y añadir nuevos mediante un pequeño formulario.

Para ello la app dispondrá de diferentes vistas (Vue-Router). Las diferentes vistas (Views) nos permitirán navegar por la SPA (Single Page Application).

Para la carga inicial de datos utilizaremos **Axios** (ya incluido en package.json) y el ficheros json (festivales.json) que está en el directorio **public**.

Este proyecto ya incluye el componente ***Inicio.vue***, y la vista ***Bienvenida.vue***, que se cargan al inicio del programa desde ***App.vue***

Tendréis que crear la vistas ***TodosFestivales.vue, FestivalesGenero.vue, AnadirFestival.vue y DetalleFestival.vue*** , para conseguir la siguiente app.

1. **Home**

Cuando se inicia la app se visualizarán todas las rutas a las que se podrá acceder haciendo click sobre cada botón. Además, en la vista Bienvenida.vue se mostrará un botón que permitirá visualizar todos los festivales.



1. **Todos los festivales (filtros y propiedades computadas)**

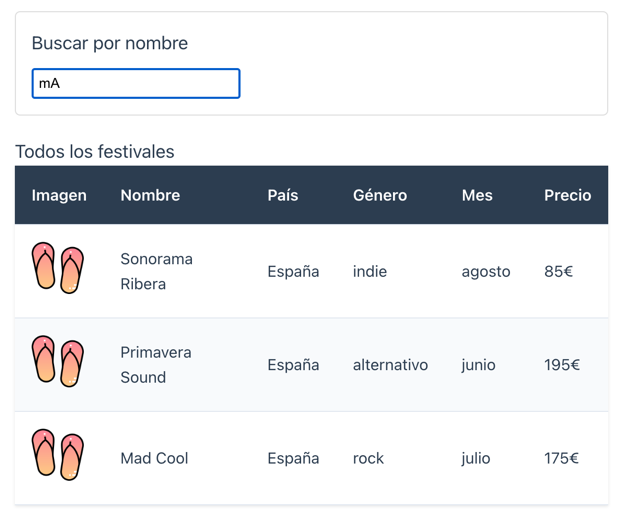
Al hacer click en el botón Todos los festivales se cargará una lista con todos los festivales que se extraerán del fichero json indicado previamente y que se encuentra en el directorio **/public/festivales.json**.



En la lista se deberá mostrar la siguiente información que se puede extraer del fichero json.

* Una imagen, cuyo nombre está indicado en el fichero json, y que se encontrará en la carpeta ***/src/assets***: primavera.png, verano.png, otono.png, invierno.png.
* El nombre.
* El país.
* El género.
* Mes en que se celebra.
* Precio.
* En cada fila se mostrará también un botón adicional que nos dará más información del festival.

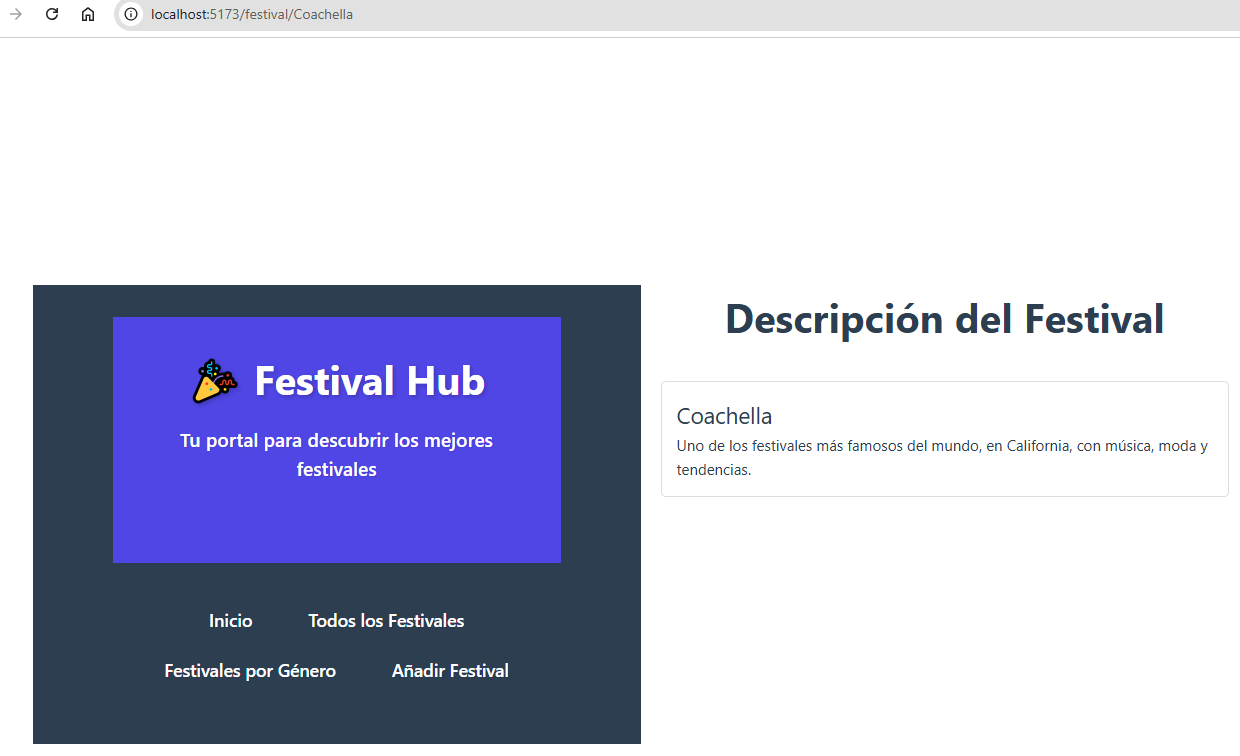
Además, hay un filtro que permitirá realizar las búsquedas sobre dicha lista, sin tener en cuenta las mayúsculas o minúsculas. Para ello se hará uso de una **propiedad computada**. Un ejemplo de funcionamiento podría ser este:



1. **Detalle del festival**

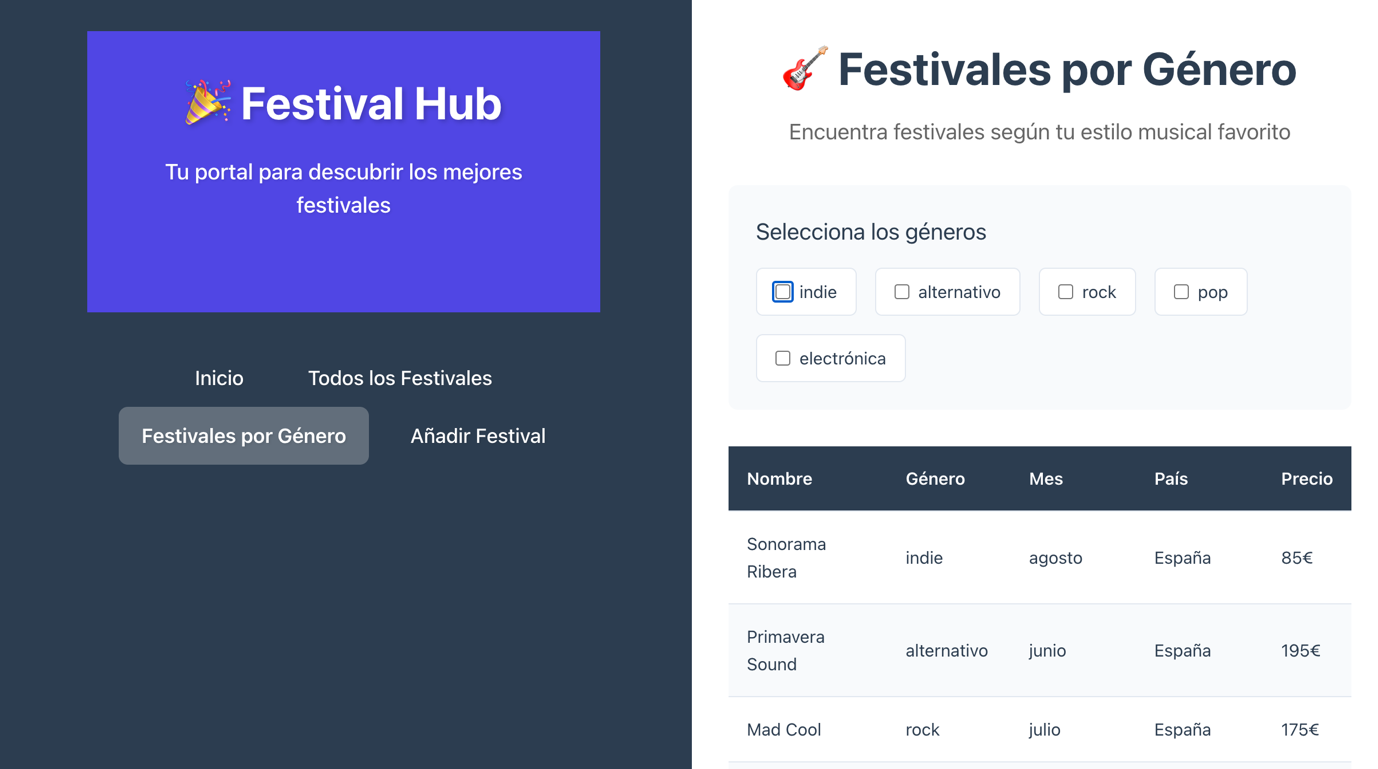
Al pinchar en el botón de +info, nos mostrará el campo descripción de ese festival. La vista en la que se muestra es **DetalleFestival.vue,** y esta vista se muestra en una ruta con parámetro **/festival/nombreFestival** Teniendo en cuenta que el parámetro será el nombre del festival.

Un ejemplo del funcionamiento de esta funcionalidad sería:



1. **Festivales por género (filtros y propiedades computadas)**

Al hacer click en el botón Festivales por género se cargará una lista con todos los festivales que se extraerán del fichero json indicado previamente, similar a la anterior pero sin imagen:



Esta página dispondrá de una serie de casillas de verificación (checkboxes) con la lista de géneros (sin repetir) filtrados a partir de la lista de festivales (**los checkboxes no deben estar hardcodeados**).

Es decir, no podrá haber un género “punk” si no hay festivales cuyo género sea “punk” del mismo modo que si se añade un nuevo festival cuyo género es “punk” deberá aparecer en la lista de géneros.

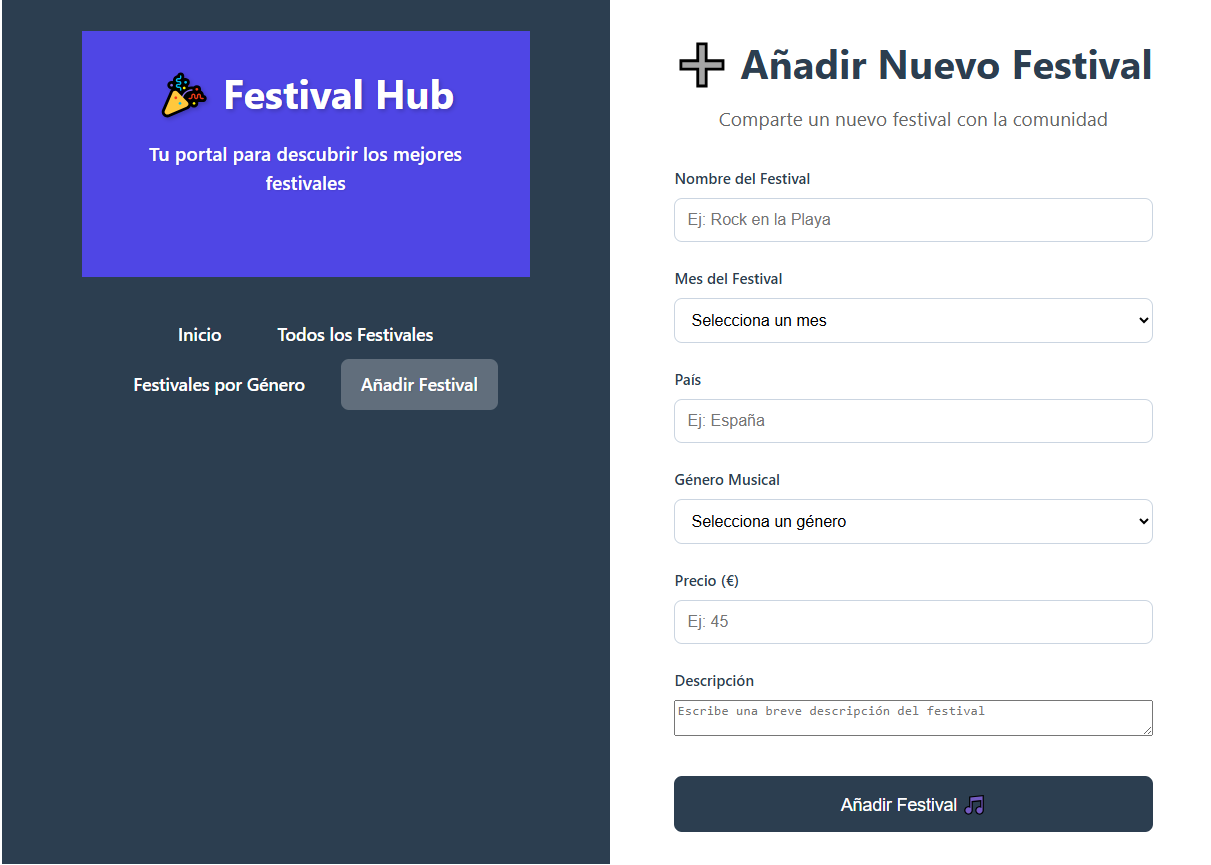
Al marcar los diferentes checkboxes se realizará el filtrado, de manera que el resultado serán los festivales correspondientes a todos los géneros marcados.

Un ejemplo podría ser el siguiente:



1. **Añadir nuevo festival (Pinia)**

Al hacer click en el botón Añadir nuevo festival se visualizará un formulario como el siguiente:



El formulario está formado por varios campos.

* Nombre y país son campos de texto.
* Precio es un campo numérico.
* Mes es un campo desplegable que se carga a partir de un array.
* Género musical, al igual que en el ejercicio anterior, es un campo desplegable que se extraerá de la lista de géneros (sin repetir) filtrados a partir de la lista de festivales hasta el momento. Es decir, no podrá haber un género “punk” si no hay festivales cuyo género sea “punk” del mismo modo que si se añade un nuevo festival cuyo género es “punk” deberá aparecer en la lista de géneros.

Cuando el usuario pulsa el botón “Añadir festival”:

* Se añade un nuevo festival al store de festivales que contiene la información del formulario.
* También se añade como campo la imagen, que será diferente en función del mes del festival (diciembre, enero, febrero y marzo será la imagen invierno.png; abril, mayo y junio será la imagen primavera.png, etc.).
* Se resetea el formulario.
* Se añade una pequeña tarjeta debajo del formulario con la siguiente información introducida:



|  |  |
| --- | --- |
| **Total ejercicio - 2** | **8** |
| 1. **Inicio** |  |
| * 1. Se han creado las 3 rutas de navegación. | 0,15 |
| 1. **TodosFestivales.** |  |
| * 1. La carga de datos de los festivales con Axios para su uso posterior es correcta | 0,50 |
| * 1. Se añade un botón en cada fila | 0,10 |
| * 1. El filtro por nombre funciona correctamente | 0,25 |
| * 1. El filtro se ejecuta mediante una propiedad computada | 0,50 |
| 1. **DetalleFestival** |  |
| * 1. Al pinchar el botón redirecciona a la vista DetalleFestival | 0,50 |
| * 1. En la vista se muestra el campo descripción del fichero json del festival pulsado | 0,50 |
| * 1. En la url aparece festival/nombreFestival | 0,50 |
| 1. **FestivalesGenero** |  |
| * 1. Al seleccionar vista aparecen todos los festivales del json pero sin la imagen | 0,10 |
| * 1. Aparecen los checkboxes de los géneros | 0,10 |
| * 1. Los checkboxes se cargan dinámicamente (no están hardcodeados) | 0,75 |
| * 1. El filtro funciona correctamente con 1 selección | 0,15 |
| * 1. El filtro funciona con varios checkboxes seleccionados | 0,75 |
| 1. **AñadirFestival** |  |
| * 1. La ruta funciona correctamente y se carga el formulario | 0,10 |
| * 1. Los meses se cargan desde un array de meses | 0,25 |
| * 1. Los géneros musicales se cargan desde el json (no están hardcodeados) | 0,50 |
| * 1. Debajo del formulario se muestra la tarjeta con la info introducida | 0,25 |
| * 1. El formulario se resetea | 0,15 |
| * 1. La pulsar insertar se añaden los datos | 0,25 |
| * 1. Se añade el campo con el nombre de la imagen según el mes | 0,40 |
| * 1. Hay persistencia de datos. Al pulsar en **TodosFestivales** aparecen los nuevos festivales introducidos. | 1,25 |
|  |  |