

Desarrollo Web en Entorno Cliente

Entregable 2^a

Evaluación - Parte 3

Actualizado Diciembre 2020

Licencia





Reconocimiento – NoComercial - CompartirIgual (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 **Importante**

 **Atención**

 **Interesante**

ÍNDICE DE CONTENIDO

1. Actividades	3
2. Autores (en orden alfabético)	4

ENTREGABLE 2ª EVALUACIÓN - PARTE 2

! **Atención:** intentad que el programa sea lo más claro y estructurado posible, así como incluir comentarios en sitios clave para facilitar la comprensión del mismo.

! **Atención:** Para considerar un ejercicio realizado, no basta únicamente con entregar el código. El alumno ha de SER CAPAZ de defender su ejercicio a petición del profesor y SER CAPAZ de realizar pequeñas modificaciones relacionadas con el mismo, con el fin de demostrar la adquisición de conocimiento y evitar cualquier sospecha de copia.

La copia se castiga con el suspenso del módulo completo.

1. ACTIVIDADES

Plazo de entrega: hasta el 22 de febrero de 2021 a las 12:00.

Peso de la nota: 10% de la nota total (Total nota entregables 35% de la nota).

Aplicación con Quasar

Este ejercicio deberá realizarse usando Quasar (con Vue y Axios para consultas AJAX), de forma que pueda generarse una aplicación web, una de escritorio y otra para Android.

Para este primer ejercicio utilizaremos la API del periódico New York Times. Para usarla necesitaremos una API KEY que obtendremos registrándose de manera gratuita:

<https://developer.nytimes.com/>

Tras registrarnos, deberemos **LEER ATENTAMENTE EL FAQ** de uso de la API.

Si tenéis problemas en el registro o dudas en general del funcionamiento de la API podéis plantear vuestros problemas en el foro.

En concreto, para el ejercicio utilizaremos la API de búsqueda de artículos

<https://developer.nytimes.com/docs/articlesearch-product/1/types/Article>

Nuestra página deberá tener los siguientes elementos Quasar:

- Un input de tipo “text” donde podamos escribir una búsqueda.
 - <https://quasar.dev/vue-components/input#Input-types>
- Dos input de tipo “text”, uno para indicar “Fecha mínima de la noticia” y otro para “Fecha máxima”. El formato será YYYYMMDD.
 - <https://quasar.dev/vue-components/input#Input-types>
- Un combo box que tenga dos opciones “primero más modernas” y “primero más antiguas”.
 - <https://quasar.dev/vue-components/select>

- Un botón “Realizar búsqueda”.
 - <https://quasar.dev/vue-components/button>
- **NOTA:** estos componentes son una recomendación. Se permiten usar otros componentes Vue presentes en “Quasar” siempre que se justifique su usabilidad.

Al pulsar “realizar búsqueda”, se realizará una búsqueda mediante AJAX mediante la API del “New York Times” con noticias que incluyan la búsqueda, entre las dos fechas indicadas y ordenadas según lo que diga el combo box (más nuevas primero, más viejas primero).

Al realizar la búsqueda será obligatorio comprobar:

- Que todos los campos están rellenos.
- Que ambas fechas tengan correctamente el formato **YYYYMMDD**. Comprobar que sean 8 números (no es necesario comprobar si el día es válido, por ejemplo si alguno pone 20001340 es problema del usuario y no debemos comprobarlo).
- Que la fecha mínima sea menor que la máxima.

La respuesta de la petición se cargará dinámicamente en una lista de items <https://quasar.dev/vue-components/list-and-list-items>

Al realizar una nueva petición AJAX, el resultado de la anterior se eliminará visualizando únicamente la nueva petición.

Instrucciones del entregable

- 1) Los ejercicios de este entregable deben realizarse usando **VUE y Axios**.
- 2) El uso del repositorio debe ser el adecuado (debe ser un sistema de control versiones, no “un lugar donde subir el resultado final. No usarlo correctamente será penalizado en la nota de forma severa.
- 3) El código debe estar ampliamente comentado (no hace falta explicar que es un if un for, etc. pero si la funcionalidad, las cosas destacadas, funcionamiento funciones, etc...), incluyendo ejercicios sencillos. El no comentarlo adecuadamente será penalizado.
- 4) Al finalizar el entregable se debe enviar mediante una “tarea” en nuestro Moodle, un enlace al repositorio GitHub privado con los ejercicios realizados y permiso para la cuenta de GitHub sergarb1.

2. AUTORES (EN ORDEN ALFABÉTICO)

A continuación ofrecemos en orden alfabético el listado de autores que han hecho aportaciones a este documento:

- García Barea, Sergi