





# **EXAMEN FINAL DWEC**

# **CONTEXTUALIZACIÓN**

A partir de un API se quiere crear una página web de noticias clasificadas por categorías. Se nos proporciona el API del servidor y el diseño de la página web que se necesita aplicar la lógica necesaria para que funcione.

#### 1. EndPoints i estructures de dades necessàries:

1. Categories:

https://news.serverred.es/api/categories

Camps: ( name i path han de tindre el mateix valor)

```
name: {
    type: String,
    required: true,
    minlength: 4,
    trim: true
},
path: {
    type: String,
    required: true,
    minlength: 3,
    trim: true
}
```

## 2. <u>Articles:</u>

Método: GET <a href="https://news.serverred.es/api/articles">https://news.serverred.es/api/articles</a>

Método: Delete <a href="https://news.serverred.es/api/articles/id">https://news.serverred.es/api/articles/id</a>

Método: GET: Votar <a href="https://news.serverred.es/api/articles/id">https://news.serverred.es/api/articles/id</a>

valores a enviar para contabilizar el voto: positivo "vote" : "upVote"

negativo: "vote" : "downVote"







Camps:

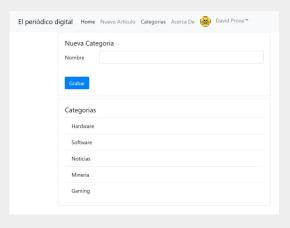
```
type: String,
    required: true,
   minlength: 5,
    trim: true
body: {
   type: String,
   required: true,
   minlength: 20,
   trim: true
author: {
    type: String,
   required: true
voteScore: {
    type: Number
deleted: {
   type: Boolean
timestamp: {
    type: Date,
   default: Date.now
category: {
type: String,
```

A la hora de crear el artículo no hay que enviar los campos timestamp y deleted. El campo category debe de ser una categoría existente en l'endpoint categories.name

## 2. Páginas a implementar

Se han de implementar 3 páginas :

1. Categorias: (Solo accesible a usuarios registrados)

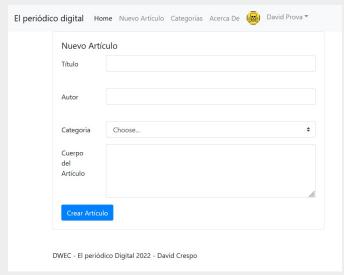




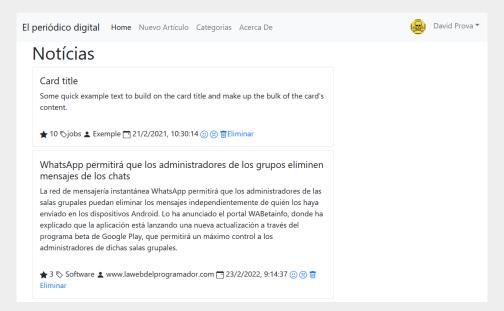




#### 2. Nuevo Artículo: (Solo accesible a usuarios registrados)



#### 3. Home / index:



En esta página se mostrarán las noticias para todos el mundo, cualquier visitante a la web podrá votar la noticia, pero solo se podrá eliminar por parte de un usuario registrado.

El registro del usuario es el mismo que hemos utilizado en las actividades anteriores. Y su endpoint es:

POST: "https://news.serverred.es/api/register"

POST: "https://news.serverred.es/api/login"

Para controlar si el usuario está registrado cabe utilizar el endpoint

GET: "https://news.serverred.es/api/areapersonal"

Se deben de ordenar los artículos mostrados de mas actuales a más antiguos.







#### Criterios de corrección

- 1. Se debe de dar de alta categorias y listar de forma automática sin refrescar la página.
- 2. Dar de alta artículos correctamente.
- 3. El select de alta de artículos debe de cargar los datos del API.
- 4. Se cargan los datos de los artículos de forma correcta en el home.
- 5. Solo se cargan los artículos que no están borrados.
- 6. Uso correcto del DOM.
- 7. Se validan correctamente los campos de los formularios.
- 8. Se gestionan los errores de validación y de respuesta del API de forma correcta.
- 9. El cálculo de los votos de los artículos es correcto.
- 10. La fecha se muestra igual que en la captura.
- 11. Se graban correctamente los datos al endpoint.
- 12. Se gestiona el flujo de los programas para reenviar a los usuarios si no estan autorizados en categorías y en alta de artículos.
- 13. Ocultar la opción de borrar en el listado de articularlos si no se esta registrado.
- 14. Ordenar el listado de artículos de mas actual a más viejo.
- 15. La aplicación tiene el aspecto que se muestra en les captures de pantalla.