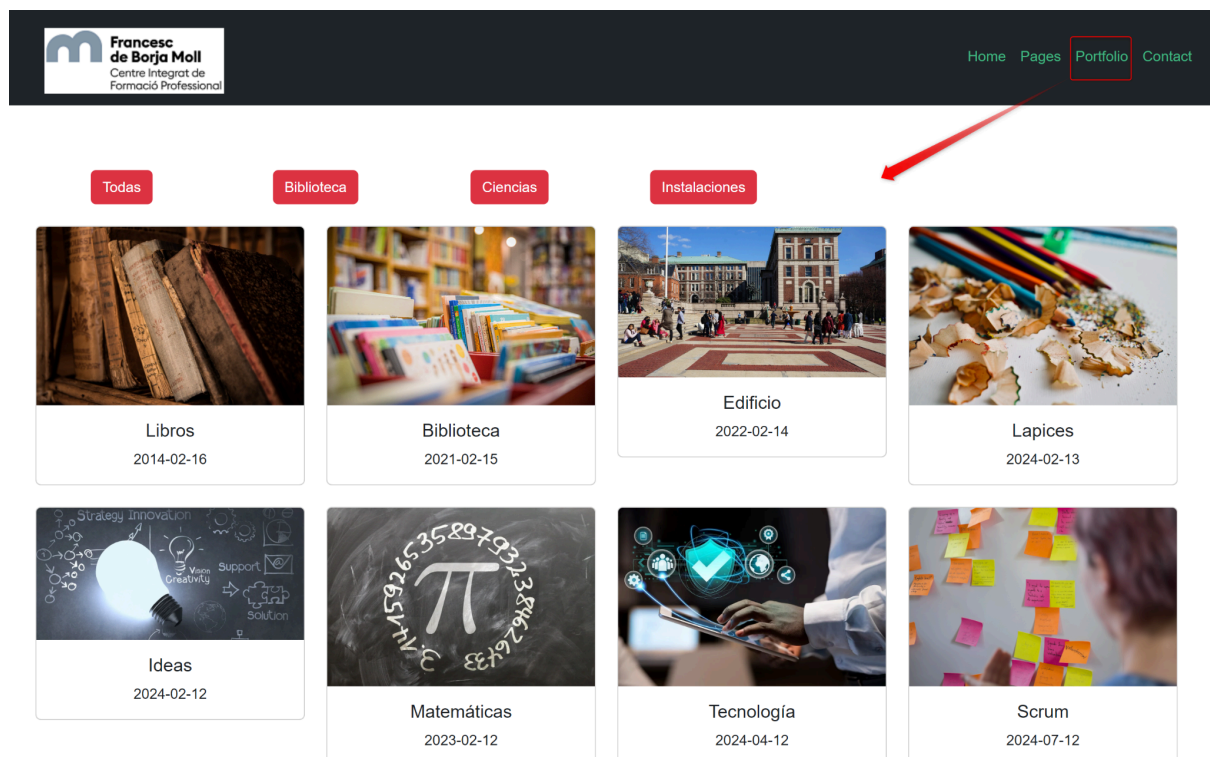


Tema 7, 8: VUE i Quasar

Part VueJS (5,75 punts)

L'empresa Borja Moll Photography ens encomana la creació d'un blog amb les fotografies més recents del Borja Moll. El dissenyador de la nostra empresa ens ha deixat una plantilla que has de descarregar i, partint d'aquesta plantilla, has de fer les següents tasques.

Aquesta part s'implementa damunt la plantilla que teniu al Repositori. Les llibreries de Vue i momentJS ja es troben importades al projecte.



Exercici 1 - Inicialització de VueJS - (0,75 punts)

Inicialitza el projecte com a aplicació de VueJS. A la vista Portfolio.vue trobareu on heu de desenvolupar la vostra part. Heu de crear 3 variables reactives que s'han de dir "entrades", "categories" i "entradesFiltrades". Les tres variables són arrays.

Exercici 2- Connexió amb el servidor remot - (1,25 punt)

Inicialitzeu les variables reactives automàticament tan bon punt s'hagi acabat de carregar la pàgina. Les variables reactives inicialitzades anteriorment han de contenir el següent:

- entrades: contingut del servidor de la funció getGaleria (veure API)
- categories: contingut del servidor de la funció getCategories (veure API)
- entradesFiltrades: una vegada que "entrades" tingui valor (i només quan entrades tingui valor), entradesFiltrades ha d'inicialitzar-se amb el mateix contingut que entrades, és a dir, "entradesFiltrades = entrades".

Exercici 3- Variables i Atributs - (1 punt)

Creeu TOTES les entrades que hi ha dins la variable "entradesFiltrades". Per a cada entrada, heu de posar:

- URL de la foto: atribut "src" de la imatge amb class="card-img-top"
- Títol de la foto: al tag h5 amb class="card-title"
- Data: al <p> amb class="card-text"

Exercici 4 - Events- (1,25 punt)

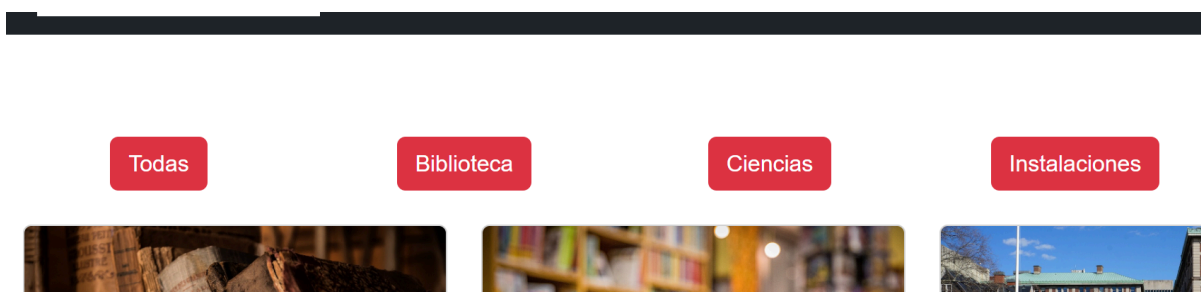
Creeu un botó que ordeni les entrades per data descendent.

Nota: la llibreria "momentJS" està ja importada per si la voleu fer servir.

Exercici 5 - Filtres - (1,5 punt)

Creeu les categories amb el contingut de la variable "categories" i afegir un nou element que digui "Totes". Cada vegada que fem clic damunt una categoria s'ha de cridar a la funció "filtrar" la qual ha de filtrar les entrades. Si fem clic a "Totes" s'ha de cridar també la funció "filtrar" i s'han de veure totes les categories.

PISTA: no modifiqueu MAI la variable "entrades", modifiqueu sempre "entradesFiltrades".





API Portfolio:

Per a cridar a l'API de l'examen heu de cridar amb els següents paràmetres:

getGaleria

URL: <http://34.90.153.139/examen2/getgaleria.php>

method: GET

Paràmetres:

Retorn: JSON amb les entrades de la galeria.

getCategories

URL: <http://34.90.153.139/examen2/getcategorias.php>

method: GET

Paràmetres:

Retorn: JSON amb un array de noms de categories.

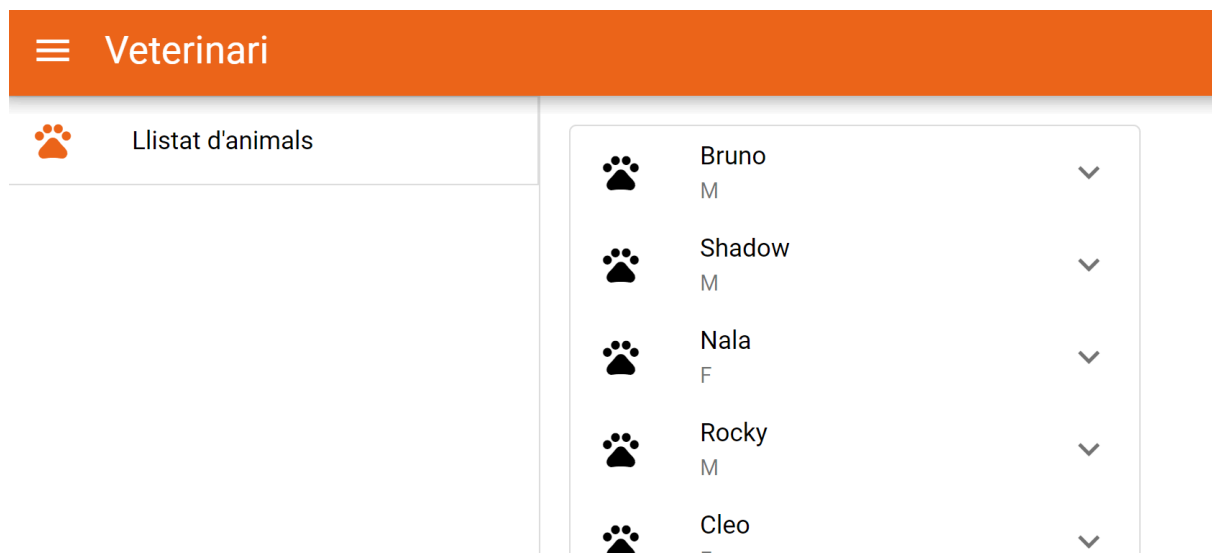
Part Quasar (4,25 punts)

L'empresa Borja Moll Veterinari ens encomana la creació d'una aplicació per gestionar una clínica veterinària. El nostre equip de desenvolupament va fer una primera part d'aquest projecte que podeu descarregar del repositori. Ens demanen fer els següents canvis.

Exercici 1 - Inicialització de Quasar- (0,75 punts)

Inicialitza el projecte com a aplicació de Quasar. Ja teniu feta la part de llistar els animals de la clínica veterinària, però el client ha demanat canviar la disposició del menú, i el color principal de l'aplicació. Heu de fer canvis pertinents per:

- Tenir el color primari com a #ef651a
- El menú s'ha de desplegar a baix del header, tal com apareix a la imatge.



Exercici 2 - Routing - (1,5 punts)

Has d'emprar l'entrada del menú "Llistat d'animals" per carregar a la dreta, al q-page-container del layout, el llistat d'animals que trobaràs a la pàgina ja existent "Animales.vue"

Has de crear nova entrada al menú que es digui Nou animal. Aquesta entrada ha de carregar una nova pàgina que hauràs de crear anomenada Animal.vue



Exercici 3 - Components - (1,5 punts)

Crea un formulari per guardar un nou animal. El formulari ha de contenir la següent relació amb els components Quasar:

Nom: QInput

Sexe: QRadio amb 2 opcions: "M, F"

Número registre: QInput

Tipus: QSelect amb 3 opcions: "Perro, Gato, Caballo"

Desar: QBtn

Quan clicam en Desar s'ha de guardar l'animal al servidor. Mirar la documentació de l'API per saber com fer la crida al servidor.

Nou animal

Nom de l'animal: Bob

Sexe de l'animal: M

Número de registre: 54625

Tipus: Gato

Nom

Bob

Sexe



Masculí



Femení

Número registre

54625

Tipus

Gato



DESAR

Exercici 4 - Plugins- (0,5 punts)

Heu d'usar el plugin de notifikacions de quasar amb l'exercici anterior de guardar animals. En cas que la resposta del servidor sigui correcta (200 OK), és a dir, l'animal es guardi correctament, es mostrarà una notifikació en verd per pantalla.

En cas que l'exercici anterior no us hagi sortit podeu emprar el següent endpoint per desenvolupar l'exercici 4:

<http://34.90.153.139/examen2/prueba.php>

Amb el paràmetre ok=true retornarà un codi http 200

`http://34.90.153.139/examen2/prueba.php?ok=true`

Amb el paràmetre ok=false retornarà un codi http d'error.

`http://34.90.153.139/examen2/prueba.php?ok=false`

API Veterinari:

save

URL: `http://34.90.153.139/ejercicios/veterinario/save.php`

Mètode: POST

Paràmetres entrada: Objecte "animal" amb els camps:

id: integer. Ha de ser buit.

nombre: string

sexo: string amb els valors "M" o "F"

numeroRegistro: string

tipo: string amb els valors "gato", "perro" o "caballo"

Exemple:

`{"id": "", "nombre": "Tobby", "sexo": "M", "numeroRegistro": "123456", "tipo": "Perro"}`

Resultat: codi HTTP correcte o erroni amb el missatge d'error