



IT ACADEMY | BARCELONA ACTIVA

EJERCICI REACTUBE: PART 1

Versió bàsica

DESENVOLUPAMENT PER COMMITS

Introducció

Amb aquest exercici posarem en pràctica els conceptes apresos de React.

L'objectiu és que a mesura que avances en la realització d'aquest, siguis més autònom i necessitis menys consells i ajudes. Sent capaç de desenvolupar funcionalitats la teva només una vegada que ho hakis finalitzat.

Manera de realització

És molt important que la realització del projecte es faci per commits separats.

Al començament de cada apartat apareix el nom que s'ha de posar al commit de la funcionalitat que es realitzarà. A continuació apareix la descripció de la funcionalitat que s'ha de desenvolupar, i finalment, apareixen els consells i enllaços a recursos que poden ser d'utilitat per a la seva realització.

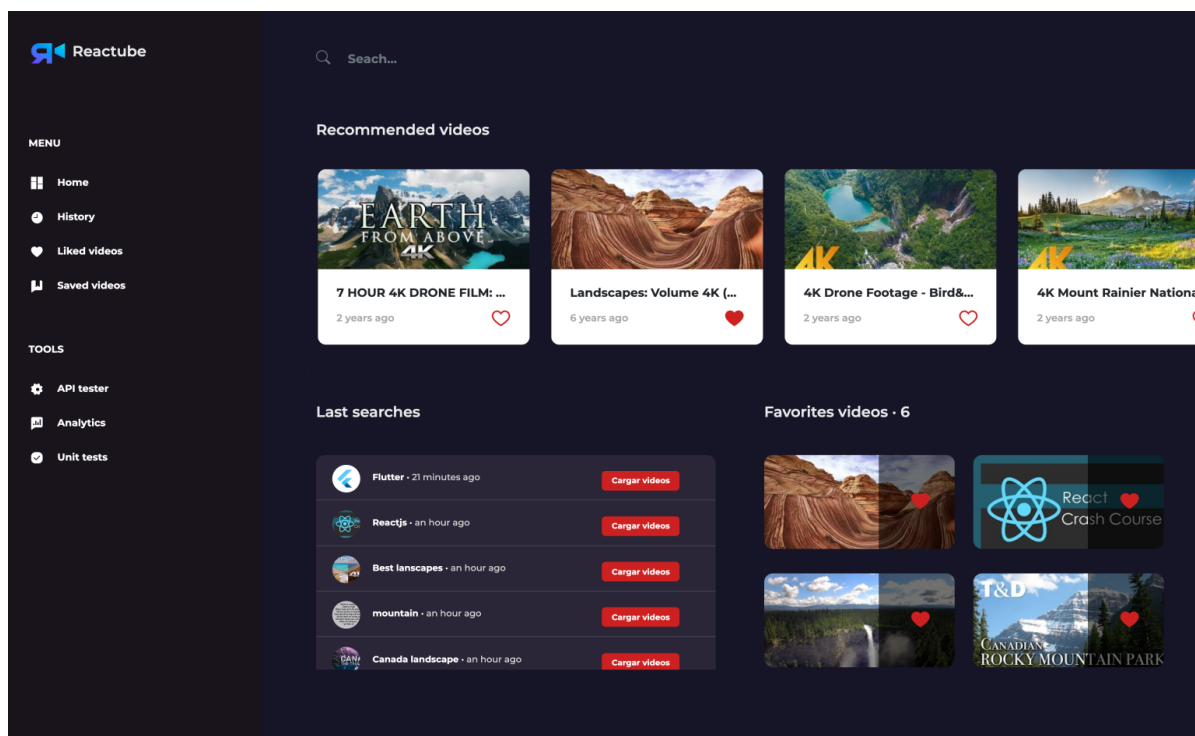
Descripció del projecte

Farem la nostra pròpia versió de youtube, consumint vídeos reals mitjançant la API que proporciona google:

<https://developers.google.com/youtube/v3/docs?hl=es>

Abans de començar convé llegir per damunt la documentació de la API de YouTube per a veure que ofereix.

En aquesta aplicació es podran buscar vídeos, hi haurà gestió de favorits, historial de cerques i detall de vídeo amb els seus vídeos relacionats:

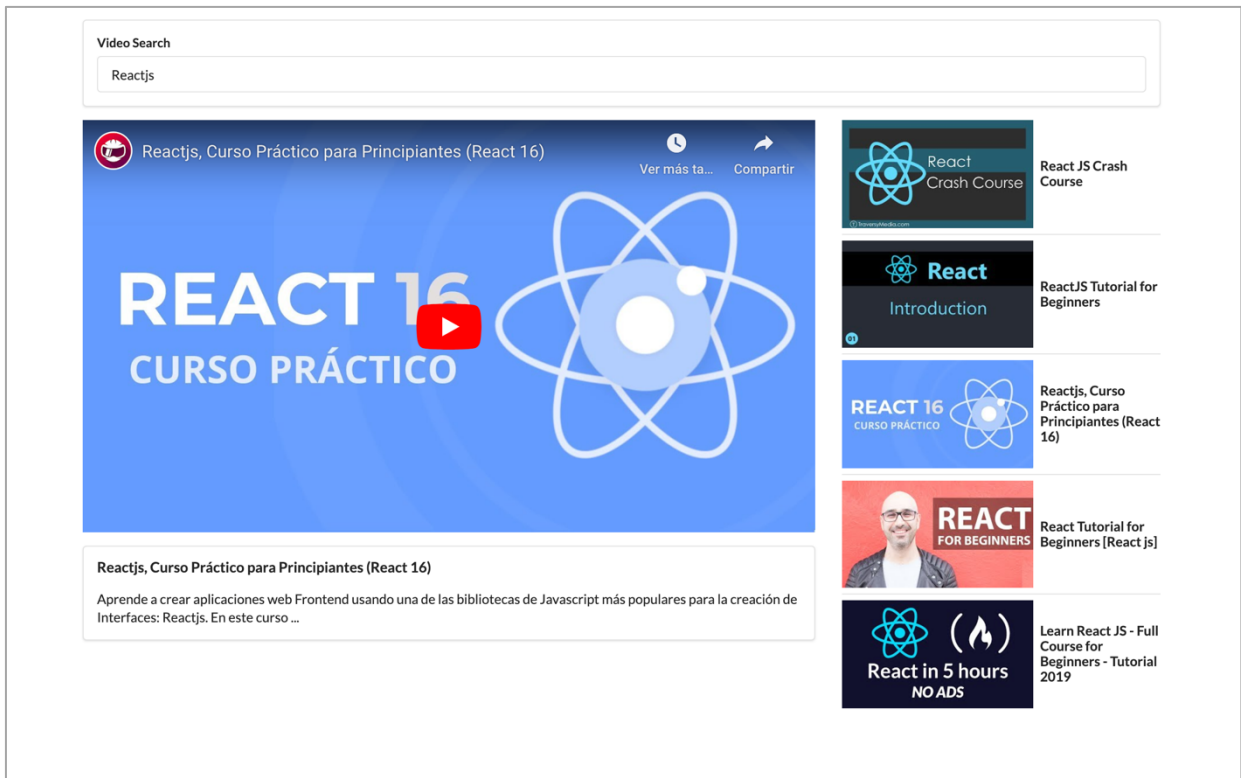


Abast d'aquest exercici

Ens trobem en la primera fase del projecte, on muntarem la base, carregant vídeos de la API de YouTube i mostrant-los per pantalla.

Més endavant, en la fase 2, modificarem la navegació de l'aplicació i implementarem funcionalitats com favorits, historial de cerca...

Aquest és l'aspecte que tindrà la nostra web una web que finalitzarem aquesta primera fase:



Context

La descripció del context està basada en «fets reals», com en nombroses ocasions hem desenvolupat els professors projectes per a clients.

Has començat a treballar en una empresa com a programador Frontend júnior. L'equip en el qual t'integraràs són 4 programadors frontend sènior, un responsable Frontend, 3 programadors Backend, el CTO (de l'anglès Chief Technical Officer) i el CEO (Chief Executive Officer).

L'empresa ha creat una plataforma educativa (tipus moodle) que ofereix a diferents clients, sent la seva principal font d'ingressos.

Com és lògic, tens por a no complir amb les expectatives posades en tu, per la qual cosa vols treballar força bé per a causar molt bona impressió.

Abans del primer dia et repasses tots els conceptes principals de javascript i de React per a estar al 100%, però les dues primeres setmanes de donen el portàtil perquè vagis instal·lant tot, vagis llegint la documentació del projecte i comencis a entendre com funciona l'equip.

Després d'aquestes dues setmanes , estàs llest per a fer grans funcionalitats. Et convoquen per a una reunió per a determinar les teves pròximes tasques i encaix en l'equip.

T'expliquen que molts clients demanen una versió «lite» del producte, una plataforma de vídeos col·laborativa, simple i sense totes les opcions extra del producte principal. A causa d'això, i al fet que tots els programadors estan molt ocupats ja que s'està llançant una nova release del producte, et demanen que duguis a terme una prova de concepte, per a poder fer una demostració als clients interessats. Al no disposar de vídeos, et demanen que consumir la API de YouTube temporalment.

Els requisits que et comuniquen d'aquesta primera fase són els següents:

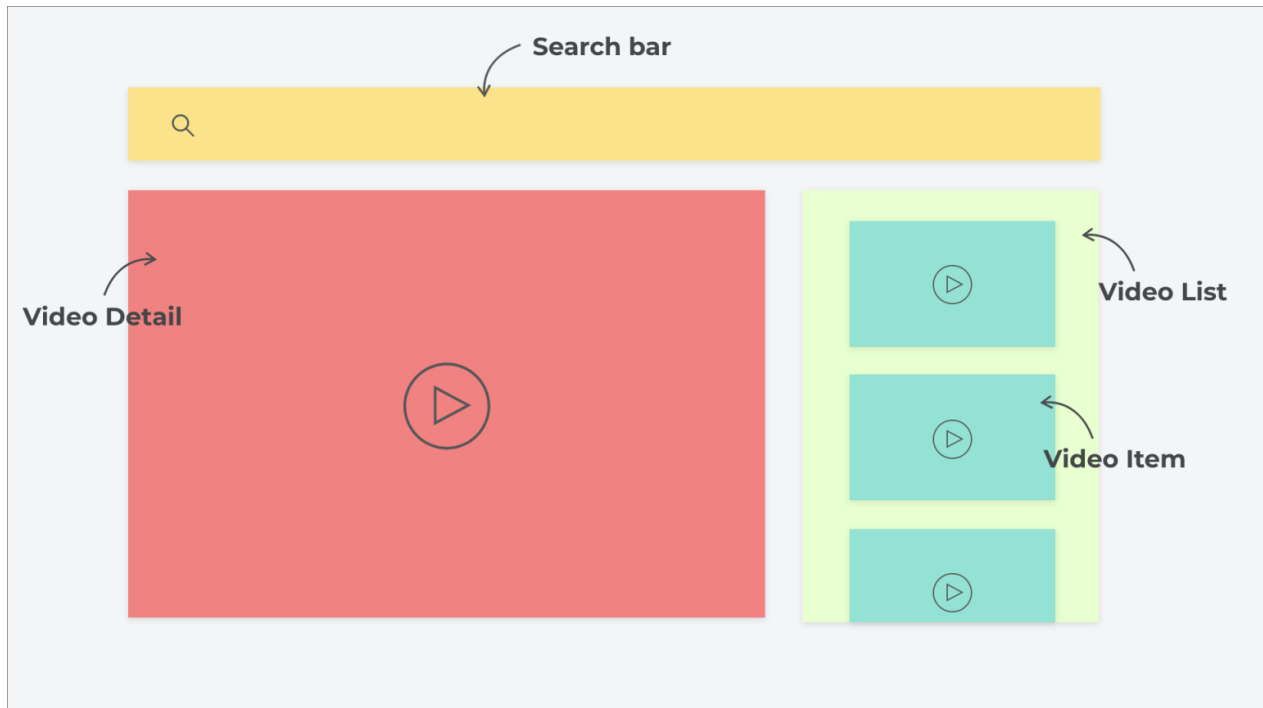
- Mostrar un cercador, que permeti a l'usuari introduir el terme de cerca.
- Quan l'usuari introdueix un terme de cerca i prem el botó buscar, es mostrarà el llistat de vídeos de la cerca realitzada, en la part dreta de la pantalla.
- Quan se selecciona un vídeo, mostrar-lo en gran per a poder visualitzar-lo, en la part esquerra de la pantalla.

Amb aquesta informació hauries de ser capaç de desenvolupar el projecte. Com portes poc temps en el món frontend, a continuació de descriuen els commits a dur a terme i es mostren ajudes de cada tasca.

Arquitectura de l'aplicació frontend

Abans de començar a programar, sempre és convenient crear un esquema dels components que integraran l'aplicació, així com un diagrama que mostri la seva jerarquia. Això ajuda a enfocar millor el projecte així com evitar possibles confusions.

Esquema components de l'aplicació:



Esquema amb el contingut de cada component:

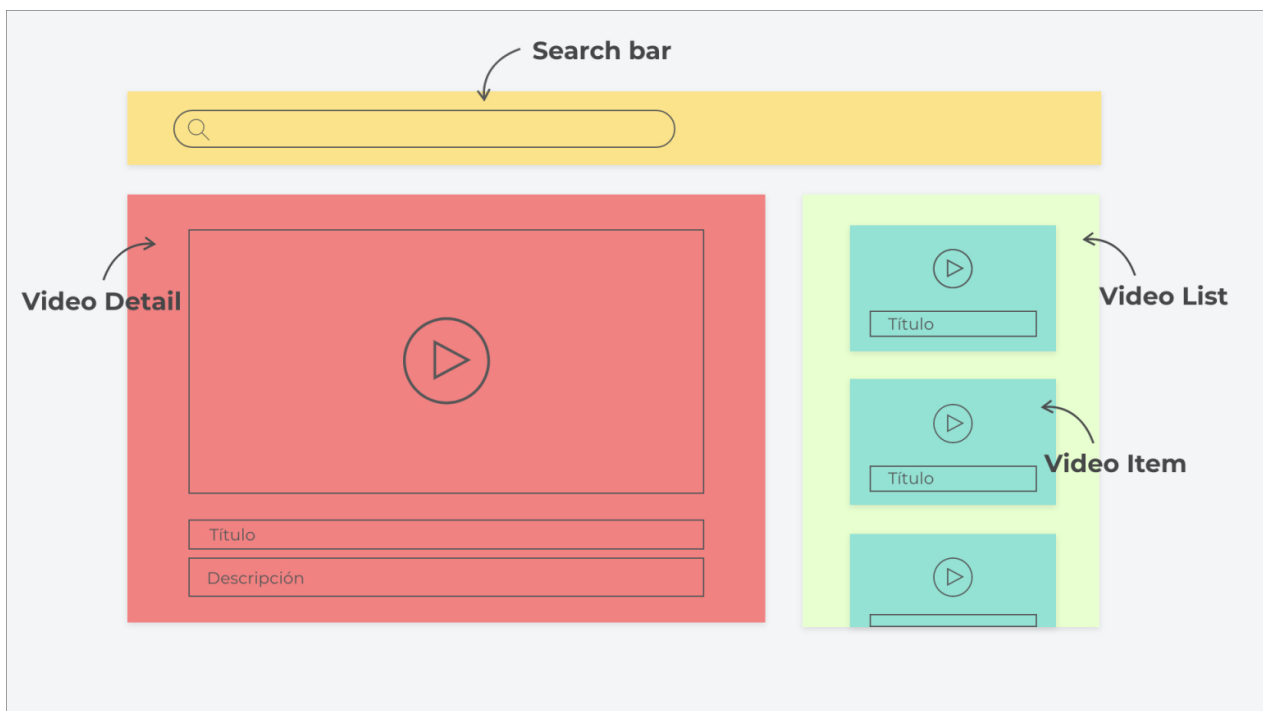


Diagrama de jerarquia de components: **a realitzar per l'alumne**

Una vegada entès el projecte i el context en el qual s'emmarca, i tenint clar els components que cal crear i la funcionalitat que busquem, és el moment de començar a programar. A continuació s'explica cada commit que s'ha de dur a terme.

Nota: En realitzar una cerca, apareixerà el llistat de vídeos en la part dreta de la pantalla. Com encara no hem seleccionat cap, la part esquerra de la pantalla apareixerà en blanc, per a evitar això pots mostrar el detall del primer vídeo del llistat.

Commit 1. Crear la base del projecte

Nom del commit: «initialCommit»

Descripció: crear l'estructura bàsica del projecte amb els paquets que creguem que utilitzarem.

Tip 1: és convenient usar el CLI create-react-app per a aquest propòsit. A més d'estalviar-nos temps, crearà una estructura de directoris estàndard.

Tip 2: és bon moment per a instal·lar els paquets que vagi a usar en el projecte. Per exemple, per a facilitar-te la crida de les APIs de recomana intalar Axios.

Tip 3: També s'ha de pensar incloure una llibreria UI per a facilitar-te la creació de la interfície (react-bootstrap, semantic UI, ant design... repassar el bloque "Estils en React i llibreries UI").

Material d'interés:

- El mòdul "Pensant en React " explica com separar una aplicació en components i els primers passos de la creació d'aquesta.

- El mòdul "Estils en React i llibreries UI " del curs conté informació útil sobre llibreries UI per a ajudar-te a decidir quin usar, i com començar a treballar amb una d'elles: react-bootstrap.

Commit 2. Component App

Nom del commit: «appComponent»

Descripció: s'ha de crear el component App, el més important de l'aplicació. Has de tenir en compte:

- Quins components inclourà (i quins són els props que proporcionarà a cadascun).
- Quins mètodes són necessaris que tingui.
- Quin és el seu state.

De moment pots fer que aquest component carregui els vídeos de YouTube i mostrar-los per consola per a saber si s'ha realitzat bé la cerca.

El array de vídeos obtinguts de la API de youtube s'han de passar per props al component videoList.

Com es mostra en l'esquema de components, aquest component App.js conté tres components: searchBar, videoList i videoDetail.

Referència de API de YouTube:

<https://developers.google.com/youtube/v3/docs>

Exemples de cridas API de Youtube:

https://developers.google.com/youtube/v3/sample_requests

Nota: Intenta resoldre la cerca de vídeos usant la API de YouTube sense l'ajuda d'un tutorial (tipus "buscar vídeos de youtube amb react i axios"), en un futur et tocarà consumir APIs pitjors documentades. És una oportunitat per a aprendre a consumir dades d'una API.

Tip 1: com hauràs pogut comprovar en molts tutorials, normalment hi ha un component que és el que conté l'estat general de l'aplicació, els mètodes principals per a dinamitzar-la així com és el que inclou a la resta de components o el routing. El component App serà aquest component principal.

Tip 2: en aquest component necessitaràs dos mètodes: «handleSubmit» i «handleVideoSelect»:

- «handleSubmit» és el mètode que s'executarà quan es realitzi una cerca, el qual acceptarà un string (. El contingut d'aquest mètode és l'anomenada a la API de youtube i el seu guardat posterior en el state.
- «handleVideoSelect» s'executarà quan se seleccioni un vídeo del llistat. El únic del que s'encarregarà és modificar l'estat.

Material d'interés:

- El mòdul "React: HTTP i API REST " conté informació de com consumir dades d'una API. També mostra dues alternatives per a fer-ho: fetch i Axios, tria la que més et convingui.

Nota: És molt important tenir molt clars els mòduls esmentats anteriorment, determinarà l'èxit i el nivell de frustració del procés de creació del projecte. Si consideres que no ho tens clar, busca recursos més estesos sobre els temes que no et sentis còmode.

Code Snippet 1. STATE:

```
state =  
{  
  videos: [],  
  selectedVideo: null  
}
```

Code Snippet 2. Crida a API youtube des de App.js quan es realitza una cerca:

```
import youtube from '../apis/youtube';  
...  
handleSubmit = async (termFromSearchBar) => {  
  ...  
  const response = await youtube.get('/search', {  
    params: {  
      q: termFromSearchBar  
    }  
  })  
  ...  
}
```

Code Snippet 3. Part del codi de apis/youtube.js:

```
...  
export default axios.create({  
  baseURL: ...,  
  params: {  
    part: ... ,  
    maxResults: ... ,  
    key: ...  
    ...  
  }  
})  
...
```


Commit 3. Component SearchBar

Nom del commit: «searchBarComponent»

Descripció: una vegada creat el component principal amb el seu estat i mètodes, anirem creant els altres components. Començarem per la barra de cerca.

Tip 1: aquest component rep com paràmetres el event handleSubmit. S'haurà d'executar handleSubmit quan es realitza una cerca. Recorda handleSubmit és un mètode del component pare.

Code Snippet 1:

```
<form onSubmit={this.handleSubmit} ...>
  <div ...>
    ...
    <input onChange={this.handleChange} value=.../>
  </div>
</form>
```

Commit 4. Component VideoList

Nom del commit: «videoListComponent»

Descripció: anem amb el següent component. Aquest component ha de mostrar llistat de vídeos que s'han carregat mitjançant la API de YouTube quan es realitza una cerca.

Tip 1: recorda, segons l'esquema de l'aplicació, que aquest component conté un altre: videoItem, el qual mostra la imatge i el nom del vídeo.

Tip 2: Quan es prem un dels vídeos del llistat s'ha de carregar aquest vídeo en videoDetail, per la qual cosa caldrà passar per props un event al component videoItem.

Tip 3: aquest component pot ser stateless, per tant, no fa falta que sigui una classe.

Code Snippet 1:

```
const VideoList = ...
...
const renderedVideos = videos.map((video) => {
  return <VideoItem ...
...
return <div ...> {renderedVideos} </div>
```

...

Commit 5. Component VideoItem

Nom del commit: «videoItemComponent»

Descripció: una vegada realitzat el llistat de vídeos, falta crear el component videoItem perquè pot renderitzar-se aquest llistat.

Ja hauries de ser capaç de realitzar aquest component, sempre seguim el mateix procés:

- Fa falta que tingui state?
- Què props o paràmetres té?
- Què és el que ha de fer aquest component?
- Conté a altres components?

Commit 6. Component VideoDetail

Nom del commit: «videoDetailComponent»

Descripció: ja només queda implementar l'últim component. Recorda't que ha de contenir el títol i la descripció del vídeo.