

ENTREGABLES

1. ENLACE A GITHUB

[sirzes02/gns_project \(github.com\)](https://github.com/sirzes02/gns_project)

2. ENLACE A YOUTUBE

[Prueba técnica GNS - Santiago Varela - YouTube](https://www.youtube.com/watch?v=...)

3. PREGUNTAS

a) ¿Qué framework o librerías utilizaste? ¿Por qué?

Durante el desarrollo de este aplicativo web se usó la biblioteca React como única biblioteca para el renderizado del entorno, la razón de la selección de esta radica en la experiencia del desarrollo de proyectos anteriores y la rapidez con la que se desempeña al momento de escribir código.

b) ¿Qué servicio(s) de Amazon Web Services utilizarías para desplegar tu aplicación?

El servicio al que se le daría uso para el despliegue, control y desarrollo frontend sería AWS Amplify.

c) ¿Qué recurso utilizaste para consumir la API de IT Bookstore? ¿Por qué?

Como biblioteca para el consumo de API se realizaron pruebas con Axios y la funcionalidad Fetch; se consume en formato JSON. Se realizaron pruebas con el aplicativo Postman y peticiones se JQuery.

d) ¿Qué librerías utilizaste y para qué funcionalidades?

- Axios 0.21.0: Para la realización de peticiones.
- Bootstrap 4.5.3: Para el estilizado de la página y como dependencia de otras bibliotecas.
- File-saver 2.0.5: Para la descarga de los archivos .CSV y .XSL.
- Match-sortes: Para verificar el emparejamiento e igualdad de búsqueda por caracteres.
- React-bootstrap: Para el uso de componentes desarrollados con anterioridad.
- React-movable: Para el control del drag and drop.
- React-table: Para el manejo de la tabla y configuraciones.
- React-toastify: Para la presentación de Toast informativos.
- Sweetalert2: Para la visualización de modales.

e) El ejercicio anterior propone una o varias funcionalidades cuya responsabilidad no está del lado del FrontEnd, sino del lado del BackEnd o es compartida

1. ¿Lograste identificarla(s)? ¿Cuáles?

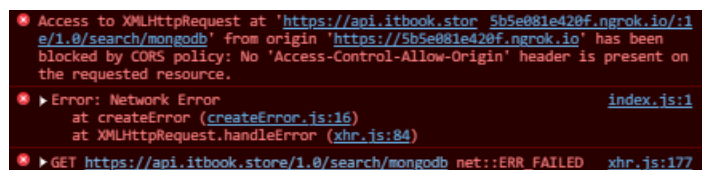
- La paginación según el consumo de la API.
- Control de cantidad de entradas visualizadas junto con el control de paginación.

II. *¿Por qué es responsabilidad del BackEnd o por qué la responsabilidad es compartida?*

- En razón al primer punto comentado en el anexo anterior, podemos definir que el proceso de paginación debe ser una responsabilidad compartida, esto, porque el manejo de las paginas deben ser indicadas desde el punto de consumo de API y de renderizado. En el caso de este proyecto, para el proceso de descargar en formato .CSV o .XSL se deben distinguir no solo paginas por petición sino la implementación de una ruta de API que retorne el valor completo de todas las búsquedas.
- Debido al segundo punto, debe ser responsabilidad compartida, al momento de cambiar la cantidad de entradas observadas en la lista, la paginación aumenta, y el control de esta paginación deja de ser referente a la petición de la página a la API.

f) *¿Qué puntos del ejercicio no lograste resolver? ¿Por qué?*

Como punto el cual no se logró desarrollar encontramos problemas con las peticiones a la API <https://api.itbook.store> (ver ilustración 1), problema de CORS; se realizaron pruebas con diferentes bibliotecas para la realización de la petición, se hicieron pruebas desde despliegues con NGROK para HTTPS.



```

Access to XMLHttpRequest at 'https://api.itbook.store/1.0/search/mongodb' from origin 'https://5b5e081e420f.ngrok.io' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.
Error: Network Error
    at createError (createError.js:16)
    at XMLHttpRequest.handleError (xhr.js:84)
GET https://api.itbook.store/1.0/search/mongodb net::ERR_FAILED xhr.js:177

```

Ilustración 1: Malfuncionamiento CORS

En ultimo recurso, se realizaron las peticiones a la API desde plataformas externas y se usó la información para el desarrollo de una sub-API con la cual interactúa la plataforma. Cabe resaltar que se tiene un manejo en el consumo de diferentes APIs pero por el error y su no solución encontrada se suprimió esta actividad.