

Avance 4

Ahora que ya se pueden visualizar los resultados de la búsqueda y cada resultado puede desplegar los detalles en el área principal de la aplicación, es necesario resolver algunos detalles como la paginación.

1. Para que los resultados de las búsquedas se presenten en páginas de hasta 10 resultados, haz lo siguiente:
 - a. Agrega y exporta al archivo de **config** la constante **RES_PER_PAGE = 10**.
 - b. En el archivo del modelo, realiza los siguientes cambios:
 - i. Importa la constante **RES_PER_PAGE**.
 - ii. En el state agrega las siguientes propiedades:
 1. Page con el valor 1 por defecto.
 2. A **resultsPerPage** asígnales el valor de la constante **RES_PER_PAGE**.
 - iii. En la parte final del archivo crea una función expresada con las siguientes características:
 1. Asígnales el nombre **getSearchResultsPage**.
 2. La función recibe el parámetro page (iguálalo a **state.search.page** para prevenir algún error si es que no se pasa nada en page).
 3. Exporta la función recién creada
 4. En el cuerpo de la función haz lo siguiente:
 - a. Asigna el valor de page pasado como parámetro a **state.search.page**.
 - b. Crea las constantes start y end, para hacer que su contenido sea dinámico, realiza lo siguiente:
 - i. start asígnales el valor de (page - 1) y multiplícalo por **state.search.resultPerPage**
 - ii. end solo multiplícalo por **state.search.resultPerPage**
 - c. Retorna una parte del arreglo **state.search.results** utilizando el método **slice** que recibirá como parámetros las constantes start y end.
 - c. En el archivo del controlador, en la función **SearchResults**, modifica la forma en la que se hace el render de los resultados para que se cambien los siguientes: en lugar de **resultsView.render(model.state.search.results)**, que ahora tome los resultados de la función **resultsView.render(model.getSearchResultsPage)**.

Avance 4

2. Ahora ya puedes visualizar que se despliegan 10 resultados por página, sin embargo, aún no tenemos los botones para avanzar y retroceder para poder navegar en las siguientes páginas, para lograrlo, realiza lo siguiente:

- a. En el directorio **views** crea un nuevo archivo paginationView.js y realiza lo siguiente:
 - i. Importa la clase View.
 - ii. Importa los iconos.
 - iii. Crea la clase PaginationView como hija de la clase Vista y dentro del cuerpo, haz lo siguiente:
 1. Declara al elemento padre como privado e indícale a través del método `querySelector` de `document` que dicho elemento padre es el que tenga como clase `pagination`.
 2. Declara la constante página actual (`curPage`) que será igual a `this._data.page`.
 3. Crea el método privado `_generateMarkup` y programa lo siguiente:
 - a. Valida cuántas páginas existen creando la constante `numPages` de la siguiente manera:

```
const numPages = Math.ceil(this._data.results.length / this._data.resultsPerPage);
```

- b. Crea los siguientes escenarios:
 - i. Estando en la página 1 y existen más páginas. En este escenario valida si la página es igual a 1 y si el número de páginas es mayor a 1. Si esa condición se cumple, regresa con template string el código html del botón con clase `pagination__btn--next` del div `pagination` (obtén el código del archivo `index.html`). Modifica el numero de la página para que el en lugar de mostrar 3, muestre `currentPage + 1`, además agrega a la etiqueta del botón el atributo `data-goto="{currentPage + 1}"`. Finalmente modifica los iconos.
 - ii. Estando en la última página, valida si `curPage` es igual al número de páginas, es `> 1`, si esa condición se cumple, regresa con template string el código html del botón con clase `pagination__btn--prev` del div `pagination` (obtén el código del archivo `index.html`). Modifica el número de la página para que el en lugar de mostrar 1, muestre la diferencia de `currentPage - 1`, además agrega a la etiqueta del botón el atributo `data-goto="{currentPage - 1}"`. Finalmente, modifica los iconos.
 - iii. Estando en cualquier página diferente a la página 1 y diferente a la última página. Valida si la página es menor al número de páginas, si eso se cumple, regresa con template strings los códigos html de los dos escenarios anteriores
 - iv. Estando en la página 1 y no existen más páginas. Es el caso por defecto y solo se regresaría una cadena vacía

Avance 4

- c. Prueba que todos los escenarios realicen correctamente su función.
- 4. Crea el método `addHandlerClick` por encima del método `_generaMarkup`. Envíale como parámetro un handler y en el cuerpo realiza lo siguiente:
 - a. Utiliza el método **`addEventListener`** de **`this._parentEl`** y pásale como parámetros:
 - i. El evento `'click'`.
 - ii. Una función anónima con el evento(e) y en el cuerpo debes utilizar la delegación de eventos y crear la constante `btn` que es igual a el elemento que se hizo clic (`e.target`), sobre este utilizar el método `closest` y pasarle como parámetro la clase del botón (`'.btn--inline'`).
 - iii. Crea la constante `goToPage` y asígnale la expresión `btn.dataset.goto`;
 - iv. Registra en la consola esta constante.
 - v. Exporta un nuevo objeto de **`PaginationView`** por defecto.
 - b. En el archivo controller realiza los siguientes cambios:
 - i. Importa la clase **`PaginationView`**.
 - ii. Dentro de la función **`controlSearchResults`**, después del render de los resultados, invoca al método `paginationView.render` y pasa como parámetro el objeto de búsqueda (`model.state.search`).

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educativo y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.