# Instrucciones de uso

Descargar fuentes desde Git: <https://github.com/kguttas/pruebamediastream.git>

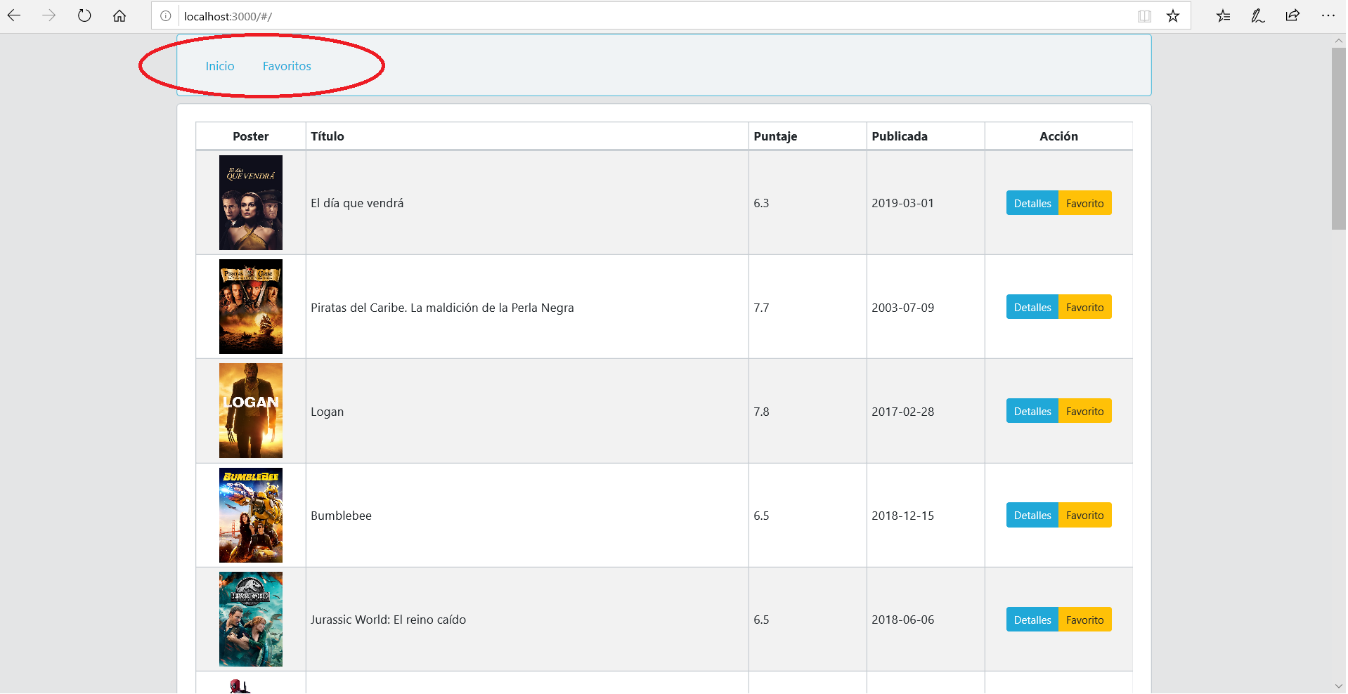
Ejecutar **npm install** en la carpeta de las fuentes.

Este se debe abrir en la ruta: <http://localhost:3000>

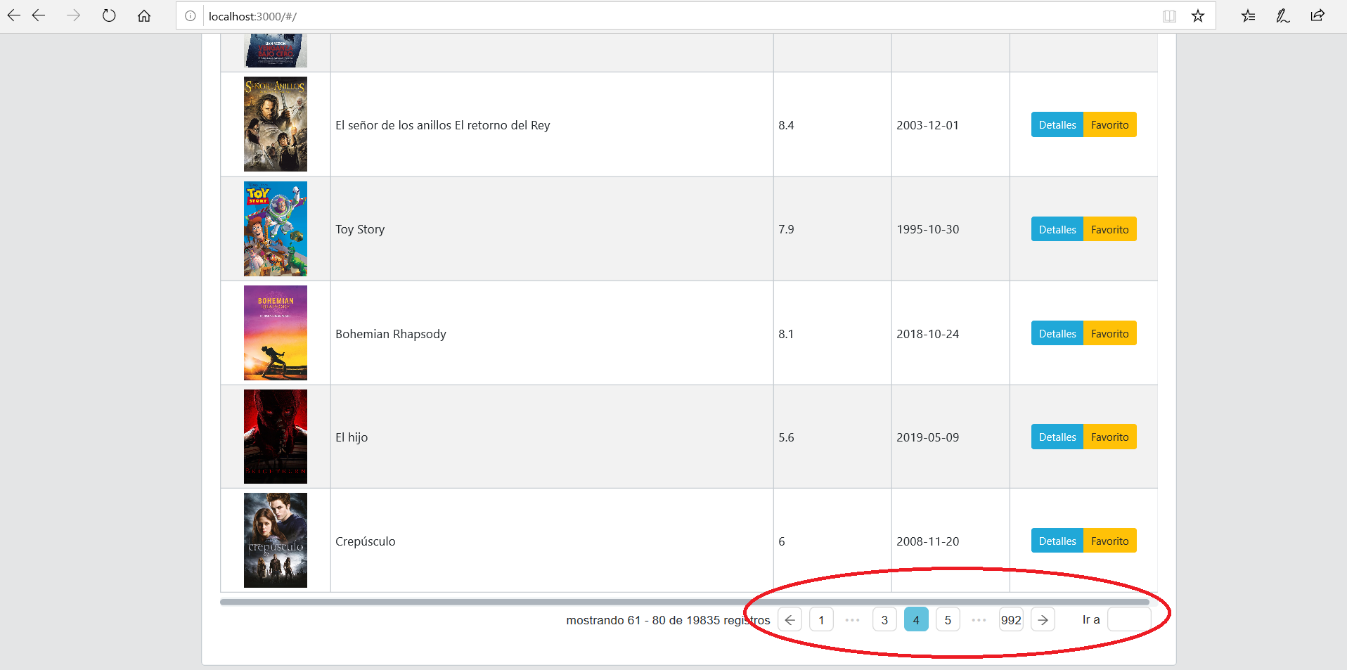
# La aplicación

Esta está dividida en tres pantallas

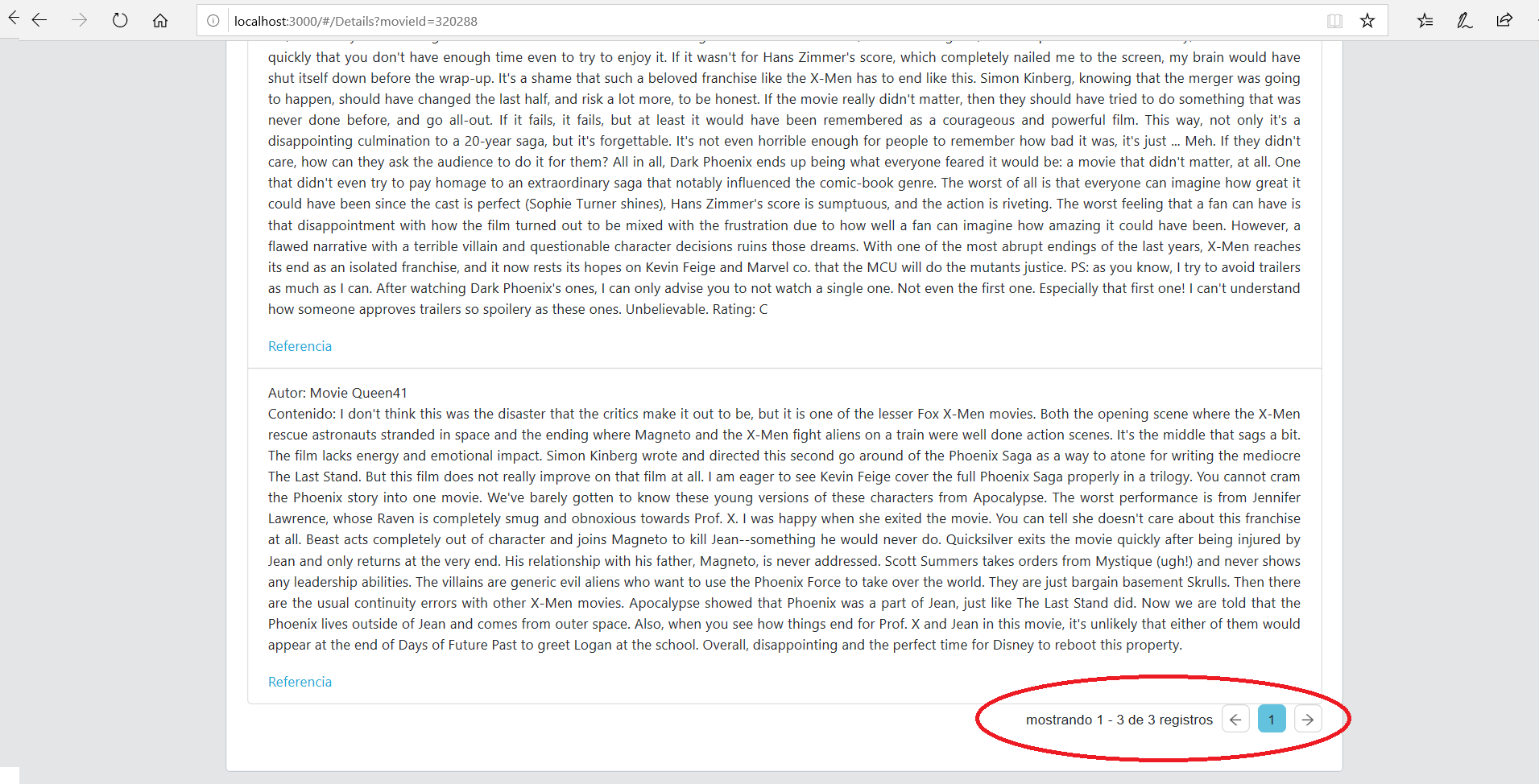
* La principal: muestra el listado de películas más populares
* La vista de detalles
* Y la vista de favoritas



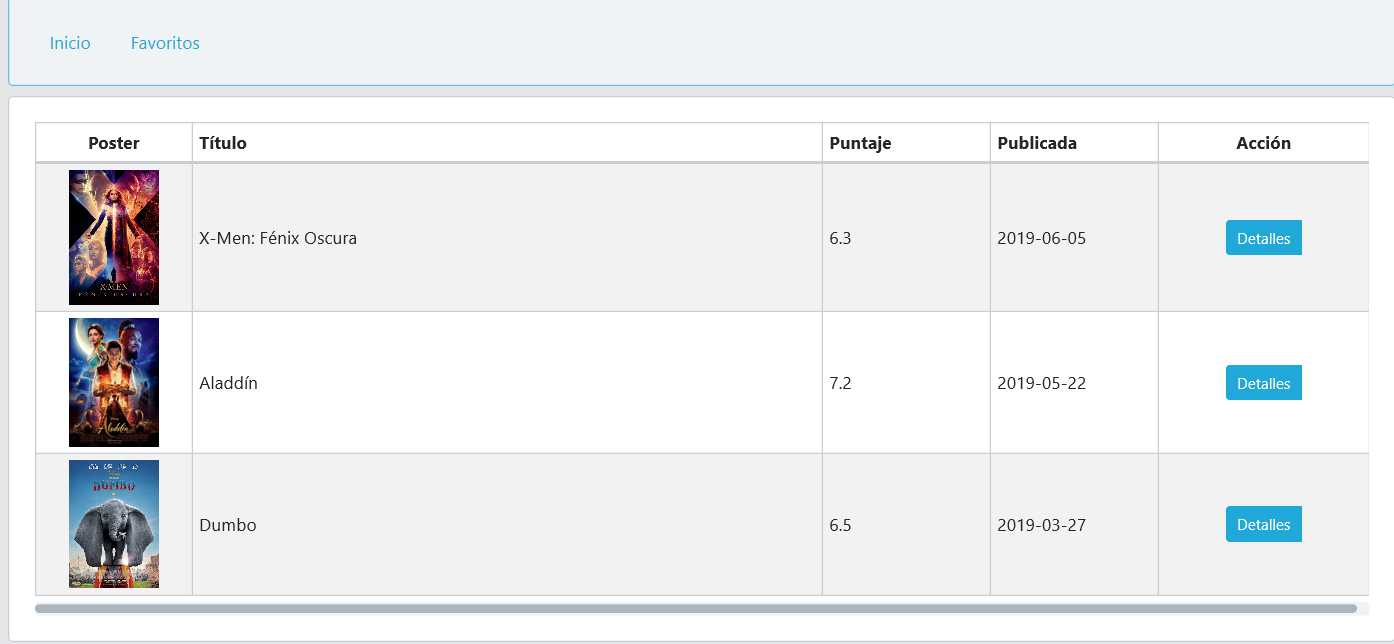
En la parte superior está el menú que permite ver la sección de Favoritos y regresar a la vista principal Home. La grilla tiene dos botones en cada fila, “Detalles”, que lleva a la vista de detalles y el botón “Favoritos” que agrega una película a favoritos.



En la parte inferior de la pantalla esta el “Paginador” de la grilla que permite ver las distintas paginas del resultado de la consulta a la API.



La vista de Detalles también tiene un “Paginador” para navegar en los resultados de los Review cuando el resultado entrega más páginas.



Desde la vista de Favoritos se puede acceder al detalle desde el botón “Detalles”

# Código de la Aplicación

La estructura del código se divide en los siguientes grupos (carpetas) principales:

* Redux: para manejar el state a nivel global se utiliza Redux, además este permite organizar las llamadas a la API en un solo archivo.
  + La carpera “actions” contiene el archivo “moviesActions.js”, que es donde encontramos las llamadas a la API mediante Axios.
  + La carpeta “reducers” contiene el archivo “moviesReducers.js”, que permite manejar la obtención de los distintos datos del state según las peticiones de los métodos “moviesActions.js” que se hacen en cada componente.
* Components:
* Favorites.jsx: componente que muestra las películas seleccionadas. Obtiene los datos desde la variable “localStorage”. El código encargado de gestionar la agregación y obtención de los datos está en el archivo “útiles.js” en la carpeta “scr”, la idea es agregar cada película solo una vez.
* Home.jsx: muestra la grilla con las películas más populares, estas se obtienen desde el método “getMoviePopular”, del archivo “moviesActions.js”, que hace uso del recurso `movie/popular?api\_key=${webConfig.apiKey}&page=${page}&language=es-CL` y son cargadas al state del componente en el método componentWillReceiveProps.
* Details.jsx: componente para la visualización del detalle de cada película. Obtiene los valores desde el método getMovieReviews para los reviews”, del archivo “moviesActions.js”, que hace uso del recuro `movie/${movieId}/reviews?api\_key=${webConfig.apiKey}&language=en-US&page=${page}`. Los datos de la película en si mismo se entregan a través de this.props.location.state, ya que estos fueron obtenido anteriormente por el componente Home.
* GridView.js: este componente se encarga de renderizar una grilla con los datos entregados por props. Esta grilla está basada en este package <https://github.com/react-bootstrap-table/react-bootstrap-table2>
* Paginator: este componente renderiza un paginador que permite consultar los datos de la API según pagina y total de registros. Este componente se basa en <https://github.com/react-component/pagination>.

El archivo “GlobalConfig.js” contiene los parámetros base de la apliación, como son el token y la URL base de la API.