

Seminario de Lenguajes

Opción: PHP, React y API Rest

2023

ReactJS

React.js, es una librería de JavaScript que se utiliza para construir interfaces de usuario. Toda aplicación web React se compone de componentes reutilizables que conforman partes de la interfaz de usuario — podemos tener un componente distinto para nuestra barra de navegación, otro para el pie de página, otro para el contenido principal, etc.

Un componente es una pieza de UI (siglas en inglés de interfaz de usuario) que tiene su propia lógica y apariencia. Un componente puede ser tan pequeño como un botón, o tan grande como toda una página.

TERCERA ENTREGA

Esta última entrega consistirá en el desarrollo de una aplicación ReactJS consumiendo los endpoints creados en la entrega anterior.

- 1) Dentro de la carpeta *src* crear una carpeta llamada *components*
Realizar los siguientes componentes dentro de la carpeta *components* con los siguientes nombres (tal como figuran en negrita)
 - a. **HeaderComponent**: el cual debe contener un logo que identifique su página (puede ser una imagen de internet o creada) y un título.
 - b. **FooterComponent** el cual debe contener los nombres de los integrantes del grupo y el año en curso
 - c. **NavBarComponent**: el cual debe permitir navegar las diferentes secciones del sitio
 - i. Home del sitio (contendrá el listado de juegos y filtro de búsqueda)
 - ii. Listado de Géneros
 - iii. Listado de Plataformas
- 2) Todas la secciones (o páginas) deben contener los siguientes componentes:
 - a. Header
 - b. Footer
 - c. Navbar para navegar la página
- 3) Dentro de la carpeta *src* crear una carpeta llamada *pages*
 - a. Crear un archivo dentro de la carpeta *pages/generos* llamado *GenerosPage*
Dicha página debe contener:

- Listado de todos los géneros
- Cada género debe tener las acciones para editar y eliminar
- Botón para crear un nuevo género

Crear un archivo dentro de *pages/generos* llamado *NewPage*

- ◆ Dicha página debe contener un formulario con un input para ingresar el nombre
- ◆ Se debe validar que el input no sea vacío antes enviarlo
- ◆ Se debe mostrar el mensaje devuelto por el endpoint

Crear un archivo dentro de *pages/generos* llamado *EditPage*

- ◆ Dicha página debe contener un formulario con un input para ingresar el nombre
- ◆ El formulario debe venir precargado con el nombre que se quiere editar
- ◆ Se debe validar que el input no sea vacío antes enviarlo
- ◆ Se debe mostrar el mensaje devuelto por el endpoint (éxito o error)

- b. Crear un archivo dentro de la carpeta *pages/plataformas* llamado *PlataformasPage*
Dicha página debe contener:

- Listado de todas las plataformas
- Cada plataforma debe tener las acciones para editar y eliminar
- Botón para crear una nueva plataforma

Crear un archivo dentro de *pages/plataforma* llamado *NewPage*

- ◆ Dicha página debe contener un formulario con un input para ingresar el nombre
- ◆ Se debe validar que el input no sea vacío antes enviarlo
- ◆ Se debe mostrar el mensaje devuelto por el endpoint

Crear un archivo dentro de *pages/plataformas* llamado *EditPage*

- ◆ Dicha página debe contener un formulario con un input para ingresar el nombre
- ◆ El formulario debe venir precargado con el nombre que se quiere editar
- ◆ Se debe validar que el input no sea vacío antes enviarlo
- ◆ Se debe mostrar el mensaje devuelto por el endpoint (éxito o error)

- c. Crear un archivo dentro de la carpeta *pages/dashboard* llamado **DashboardPage**.
Esta página deberá mostrarse como página principal (como *index.php* en la entrega uno)

Dicha página debe contener los siguientes elementos:

- Un formulario que permita filtrar el listado por nombre, género y plataforma. El listado se debe poder ordenar por nombre de manera ascendente y descendente.
- Listado de juegos

Requisitos obligatorios

- Usar la última versión estable de **nodejs**.
- El nombre del proyecto debe ser **plataforma-juegos**
- Si instalan librerías externas dejarlas documentadas en el archivo **README.md**
- Los estilos NO pueden estar embebidos en el código
- Para la creación de componentes se debe utilizar componentes de funciones (Function Components).

Metodología para la entrega en Ideas

- Hacer un archivo zip o rar con todos los archivos del proyecto
- Subir a la sección del grupo asignada
- **Fecha de entrega: 29/6**