**PRUEBA REACT JS**

Desarrolle una página web para un cinema utilizando ReactJS para el front y Express para el back. Puede utilizar las dependencias que usted requiera para el desarrollo de la prueba.

Se debe crear un back que funcionará como API para consumir toda la información que se desea mostrar en la página web. Para la parte del Front se deben crear los componentes necesarios para cumplir con los requerimientos de la prueba.

**Base de datos**:

Se debe crear una base de datos que contenga las tablas mostradas a continuación:

* Usuarios (Cedula, num\_celular, correo, contraseña)
* Películas (titulo, imagen, descripcion, duración, genero, hora)
* Salas (numero, capacidad, tipo)
* Reservas (cedula,titulo,sala,hora)

**Frontend:**

**1.1** Página inicial en donde se puedan visualizar la cartelera del cinema. (mínimo 5 películas) se debe mostrar imagen de la película y título.

**1.2** Se debe crear un menú en la parte superior que contenga los siguientes items: (El menú debe ser mostrado en todas las páginas)

* **Registro:** Al dar clic se debe dirigir a otra página en donde se muestren cuatro campos de texto para que el usuario pueda ingresar: número de cedula, numero celular, correo electrónico y contraseña. La contraseña debe tener mínimo 8 caracteres con por lo menos un número y un carácter especial, se debe validar el campo de correo para que contenga la estructura de @gmail.com. Se debe crear un botón para registrarse, al momento de dar clic se debe enviar una petición a la API que ingrese la información recolectada desde el Front en la tabla “Usuarios”. Se debe mostrar un pop up para confirmar si el registro fue exitoso o no.
* **Ingreso:** Al dar clic en ingreso, se debe desplegar dos campos y un botón “iniciar sesión” en donde se debe ingresar la cedula y la contraseña, al momento de dar clic en el botón se debe consultar en la API que el usuario este previamente registrado y que sus credenciales sean correctas, en caso de no serlo se debe mostrar un pop up con la alerta, sí son correctas se debe iniciar sesión con ese usuario.  
  Cuando se inicia sesión debe aparecer en el menú el correo del usuario, adicionalmente se debe crear una cookie con la sesión que debe ser proporcionada por la API utilizando JWT.
* **Películas:** Al dar clic en películas, se debe dirigir a otra página en donde se muestren todas las películas de la base de datos con todos sus elementos: Título de la película, imagen de la película, descripción, duración de la película, género y horas en las que presentan la película. Cuando se dé clic en alguna de las horas disponibles para la película se debe mostrar un pop up en donde se pueda confirmar si se desea reservar la película elegida (Solamente se debe mostrar si ya se ha iniciado sesión) al momento de dar clic en reservar se debe guardar en la tabla “Reservas” la cedula, el título de la película, la sala de la película (se debe elegir aleatoriamente entre las salas que estén creadas en la tabla “Salas”), y la hora elegida por el usuario. Esta pagina debe contener un filtro en forma de lista para las categorías de las películas.
* **Cerrar sesión:**  Se debe mostrar el item de cerrar sesión cuando previamente se ha iniciado sesión, al momento de dar clic en iniciar sesión se debe destruir la cookie y debe aparecer nuevamente el item de ingreso**.**
* **Admin:** Cuando se ingrese con la cedula: 111222333 y con la contraseña root, se debe mostrar en el menú el item “Admin”, al dar clic en este item se debe dirigir a otra página en donde se puedan realizar las siguientes acciones:
  + Agregar películas, se deben mostrar los campos necesarios para agregar una nueva película, debe permitir cargar una imagen.
  + Modificar películas, se debe permitir cambiar toda la información de la película seleccionada incluida la imagen previamente cargada,
  + Eliminar películas, se debe permitir seleccionar una película y eliminarla de la base de datos.
  + Agregar salas, se deben mostrar los campos necesarios para agregar una nueva sala.
  + Modificar salas, se debe permitir cambiar toda la información de la película seleccionada.
  + Eliminar salas, se debe permitir seleccionar una sala y eliminarla de la base de datos.

NOTA: Para la vista de administrador no se presenta mockup, esta vista debe realizarse de la forma que mejor le parezca.

**Backend:** Realice una API que permita:

1.0 Se deben crear las rutas o end points para consumir la API desde el front.

1.1 Consultar la tabla “Usuarios” para comparar con los datos ingresados por el usuario al momento de iniciar sesión, validar si son correctas y enviar como respuesta los datos necesarios (OK, correo, token generado con JWT).

1.2 Consultar la tabla “Películas” para la página inicial, la pestaña de películas y para la pestaña de Admin.

1.3 Ingresar un usuario nuevo a la tabla “Usuarios”

1.4 Se deben crear métodos que permitan ingresar nuevas películas, incluyendo imágenes que deben ser almacenadas en el Backend, modificar y eliminar películas previamente creadas.

1.5 Crear los métodos necesarios para cumplir con los requerimientos de la página web previamente descritos.

Es necesario hacer un documento donde se explique el funcionamiento de la aplicación y las buenas prácticas utilizadas en el desarrollo.

**Al finalizar el desarrollo cree un repositorio público en GitHub cargue todos los archivos necesarios para su funcionamiento. En un archivo txt indique los querys necesarios para crear la base de datos y alimentarla, especifique el motor de base de datos que utilizó.**

**Se darán puntos adicionales por el uso de dependencias como redux, sass, jest.**

**Enlace GitHub: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PRUEBA BASES DE DATOS.**

A continuación, cree un Trigger que permita guardar en una tabla llamada “Backup” todos los datos de las películas que fueron eliminadas. Es decir, al momento de eliminar la película desde el “Admin” se debe eliminar de la tabla “Película” pero se debe guardar automáticamente ese registro eliminado en la tabla “Backup”

**PRUEBA DIAGRAMA DE FLUJO**

La empresa “cuenta los días”, tiene 20 empleados validando fechas todos los días, el proceso consiste en analizar una fecha entregada y contar los días transcurridos desde el inicio del año hasta esa fecha. Actualmente el proceso lo realizan manualmente y con una calculadora. Ayuda a esta empresa para automatizar el proceso.

Se tiene la variable Fecha, y puede encontrarse en los siguientes formatos:

1. dd/MM/yyyy
2. yyyy-MM-dd
3. dd/XXXX/yyyy

la distribución de los formatos de fechas (100%) encontradas se distribuyen de la siguiente forma:

1. tipo 1 = 20%
2. tipo 2 = 30%
3. tipo 3 = 50%

Realice un algoritmo en **Javascript** que interprete la variable fecha y entregue el día del año correspondiente:

1. Ej: 01/01/2019, variable día = 1
2. Ej: 04/02/2019, variable dia = 35
3. Ej: 31/Dic/2019, variable dia = 365
4. **Nota**: No tome en cuenta los casos de año bisiesto.

Realice un diagrama de flujo del código desarrollado para solucionar el problema anteriormente descrito.