Nombres: Miguel Ángel Pérez – Camilo Andrés Molano – Juan Esteban Briceño

No

***Framework: React y Node.js***  
  
**Frontend React:**

React es una biblioteca de JavaScript desarrollada por Facebook que se utiliza para construir interfaces de usuario interactivas y eficientes para aplicaciones web. Aquí te proporcionaré una descripción general de sus ventajas y desventajas:

Ventajas de React:

* Virtual DOM: React utiliza un Virtual DOM para mejorar el rendimiento de la aplicación. En lugar de actualizar directamente el DOM, React actualiza primero un Virtual DOM y luego realiza una comparación eficiente para determinar los cambios necesarios en el DOM real.
* Componentización: React fomenta la creación de componentes reutilizables, lo que facilita la gestión y el mantenimiento del código. Los componentes pueden encapsular lógica y presentación, lo que mejora la modularidad del código.
* React Native: Permite el desarrollo de aplicaciones móviles utilizando la misma base de código que React, lo que facilita la transición entre el desarrollo web y móvil.
* Comunidad activa: React tiene una comunidad grande y activa, lo que significa que hay una gran cantidad de recursos, bibliotecas y herramientas disponibles. También hay una abundancia de documentación y tutoriales.
* Flexibilidad: React puede integrarse fácilmente con otras bibliotecas y frameworks. Además, puedes usarlo junto con herramientas como Redux para gestionar el estado de la aplicación de manera eficiente.

Desventajas de React:

* Curva de aprendizaje inicial: Para algunos desarrolladores, especialmente aquellos que no están familiarizados con el paradigma de programación declarativa y la sintaxis de JSX, puede haber una curva de aprendizaje inicial.
* Herramientas de desarrollo: Aunque React cuenta con un conjunto sólido de herramientas de desarrollo, algunas características avanzadas pueden requerir el uso de herramientas externas o bibliotecas.
* Toma de decisiones: La flexibilidad de React puede llevar a la toma de decisiones más compleja, ya que hay muchas formas de abordar un problema. Puede ser un desafío para los equipos nuevos o menos experimentados en la plataforma.
* SEO: Aunque React ha mejorado en términos de SEO, las aplicaciones que dependen en gran medida de la carga dinámica pueden enfrentar desafíos relacionados con el SEO. Sin embargo, hay soluciones y enfoques para abordar este problema.

En resumen, React es una opción popular y sólida para el desarrollo frontend de aplicaciones web debido a sus numerosas ventajas, pero es importante considerar las desventajas y evaluar si se adapta a las necesidades específicas de tu proyecto.  
  
**Backend Node.js:**

Node.js no es un framework en sí mismo, sino un entorno de ejecución para JavaScript en el lado del servidor. Sin embargo, es comúnmente utilizado como base para construir aplicaciones del lado del servidor. Aquí tienes una descripción general de Node.js, junto con sus ventajas y desventajas:

Node.js es un entorno de ejecución de JavaScript del lado del servidor basado en el motor V8 de Google. Permite a los desarrolladores utilizar JavaScript para escribir scripts del lado del servidor, lo que antes solo era posible en el lado del cliente. Además, utiliza un modelo de entrada/salida no bloqueante y orientado a eventos, lo que lo hace eficiente y escalable.

Ventajas de Node.js:

* Velocidad y rendimiento: Node.js es conocido por su rendimiento rápido debido a su modelo de entrada/salida no bloqueante y al motor V8 de Google. Es especialmente eficiente para aplicaciones en tiempo real y procesamiento de eventos.
* JavaScript en todo el stack: Permite a los desarrolladores utilizar JavaScript tanto en el lado del cliente como en el lado del servidor, lo que facilita la transición de datos entre ambas partes y mejora la coherencia del código.
* NPM (Node Package Manager): Node.js viene con NPM, un sistema de gestión de paquetes que facilita la instalación, gestión y compartición de bibliotecas y herramientas de terceros.
* Comunidad activa: Al igual que React, Node.js tiene una comunidad activa y vibrante. Hay una gran cantidad de módulos y paquetes disponibles a través de NPM, y la comunidad proporciona amplia documentación y soporte.
* Escalabilidad: La arquitectura no bloqueante de Node.js y su capacidad para manejar múltiples conexiones simultáneamente lo hacen altamente escalable, lo que es beneficioso para aplicaciones que necesitan manejar muchas conexiones concurrentes, como aplicaciones en tiempo real.

Desventajas de Node.js:

* No es ideal para tareas intensivas en CPU: Node.js no es la mejor opción para tareas que requieren una gran cantidad de procesamiento en CPU, ya que su rendimiento está más orientado a operaciones de entrada/salida.
* Madurez del ecosistema: Algunas bibliotecas y herramientas pueden no ser tan maduras como las de lenguajes más establecidos en el lado del servidor, aunque esto está cambiando a medida que la comunidad de Node.js sigue creciendo.
* Callback hell: Aunque las promesas y async/await han mejorado la situación, el anidamiento excesivo de callbacks en código asincrónico (callback hell) puede dificultar la lectura y mantenimiento del código.
* Manejo de errores: El manejo de errores puede ser un desafío en aplicaciones grandes debido a la asincronía de Node.js. Es crucial tener un buen manejo de errores para evitar problemas en producción.

En resumen, Node.js es una elección sólida para el desarrollo del lado del servidor, especialmente para aplicaciones en tiempo real y basadas en eventos. Sin embargo, es esencial evaluar las necesidades específicas de tu proyecto y considerar las desventajas antes de tomar una decisión.

***Propuesta del proyecto:***

El proyecto va dirigido a un gimnasio o a influencers del fitness en el cual el cliente del proyecto o algún usuario con el permiso adecuado (rol) puede agregar ejercicios a la página o asignar roles a usuarios. Los usuarios pueden escoger una lista de ejercicios los cuales los pueden agregar a una lista de 7 días, es decir, el usuario puede crear su propia rutina de entrenamiento con los ejercicios proporcionados por el gimnasio o el influencer. Los ejercicios tienen información multimedia para que el usuario realice el ejercicio correctamente.  
  
en base a este proyecto:

El proyecto va dirigido a un gimnasio o a influencers del fitness en el cual el cliente del proyecto o algún usuario con el permiso adecuado (rol) puede agregar ejercicios a la página o asignar roles a usuarios. Los usuarios pueden escoger una lista de ejercicios los cuales los pueden agregar a una lista de 7 días, es decir, el usuario puede crear su propia rutina de entrenamiento con los ejercicios proporcionados por el gimnasio o el influencer. Los ejercicios tienen información multimedia para que el usuario realice el ejercicio correctamente.

los roles predeterminados van de la siguiente manera: Admin, secretaria, premium, registrado, visitante. el perimer usuario creado tiene el rol de admin.

El admin puede dar roles a usuarios, agregar ejercicios, crear rutinas y ver los ejercicios por completo (nombre, video, foto, descripción, etc).

Secretaria puede todo menos dar roles.

Premium (usuario que paga) puede hacer todo menos dar roles y agregar ejercicios.

Registrado solo puede ver la lista de ejercicios (solo los nombres de los ejercicios).

Visitante no puede hacer nada.

el funcionamiento de la pagina es el siguiente:

el administrador o la secretaria puede agregar ejercicios a la pagina para que los otros usuarios los puedan ver. cuando un usuario nuevo(rol visitante) visita la pagina se va encontrar con un apartado de registrarse o loguearse localizado arriba a la derecha. en la página principal va a estar un cuerpo humano con vista frontal y trasera el cual se le ven delimitados cada musculo. un usuario con permiso de agregar ejercicios tiene un apartado para agregar ejercicios en su dashboard en el cual al agregar un ejercicio tiene lo siguiente:

* Nombre del ejercicio.
* Ubicación del ejercicio (es decir, el o los músculos que trabaja dicho ejercicio pueden ser más de uno)
* Foto o video del ejercicio.
* Descripción del ejercicio.
* Calificación del ejercicio (es decir que tan buen ejercicio es del 1 al 10 en cuanto a los músculos que está trabajando.

Un usuario nuevo (rol visitante) cuando le da clic a un musculo del cuerpo humano mencionado anteriormente le saldrá una alerta que le muestra que primero debe loguearse o registrarse para ver los ejercicios. Si un usuario ya está registrado (rol registrado) le va a aparecer una opción de suscribirse localizado arriba a la derecha y si le da clic a un musculo del cuerpo humano mencionado anteriormente, le van a salir todos los ejercicios que están relacionados con ese musculo en el orden de su clasificación del ejercicio y le da clic a algún ejercicio le saldrá un mensaje que dice que primero se tiene que suscribir.

Un usuario suscrito (rol premium) puede ver por completo un ejercicio (nombre, video, foto, descripción, etc). También se le va a habilitar un apartado en su dashboard que dice “rutinas” el cual puede crear una rutina y ver las rutinas creadas. Cuando el usuario crea una rutina le sale lo siguiente:

* Opción para ponerle nombre a la rutina.
* una tabla con los días de la semana y en cada día le sale un botón que dice agregar ejercicio. Cuando se le da clic al botón le sale lo siguiente:

1. cuerpo humano con vista frontal y trasera el cual se le ven delimitados cada musculo. Si le da clic a un musculo del cuerpo humano mencionado anteriormente, le van a salir todos los ejercicios que están relacionados con ese musculo en el orden de su clasificación del ejercicio, en cada ejercicio van a parecer dos botones uno que dice agregar y otro que dice ver. Si se le da clic al de agregar, lo va añadir al día de la semana en la que se encontraba en un principio (pueden haber muchos ejercicios en un solo día). Si se le da clic al de botón “ver” le mostrará por completo el ejercicio (nombre, video, foto, descripción, etc).
2. Un botón de “aceptar” el cual lo regresa al apartado donde se muestra la tabla de los días de la semana con los ejercicios que agregó (si se añadieron añgunos).
3. Un botón de “cancelar” el cual lo regresa al apartado donde se muestra la tabla de los días de la semana pero sin ningún cambio.

* Si el usuario agregó ejercicios de a un día de la semana va a aparecer la tabla de los días de la semana y debajo de cada día (columna) van a aparecer la lista de ejercicios que agregó y con un botona justo al lado del ejercicio que dice “quitar”. Si le da al botón quitar, lo elimina de dicha lista
* Un botón que dice “aceptar” el cual lo regresa al apartado donde se muestra el agregar rutinas con la rutina creada.
* Un botón que dice “cancelar” el cual lo regresa al apartado donde se muestra el agregar rutinas pero sin agregar la rutina que estaba creando.

Usuario -> lista de rutinas -> lista de días 1 -> lista de ejercicios