

EVALUACIÓN EXTRAORDINARIA

Desarrollo Web en entorno cliente
CFGS DAW

Álvaro Maceda Arranz

a.macedaarranz@edu.gva.es

2023/2024

Versión:240511.1921

Licencia



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

CONTENIDO

1. Enunciado.....	3
1.1 Ventana de chat.....	3
Banner inicial.....	4
Redimensión de la ventana de texto.....	4
1.2 Lista de conversaciones.....	4
2. Servidor de chat.....	4
3. Linter.....	5
4. Entrega.....	5
5. Criterios de evaluación.....	5

EXAMEN EVALUACIÓN ORDINARIA

Revisa los criterios de evaluación para saber que es lo que debes tener en cuenta a la hora de realizar el ejercicio: **no es suficiente que el programa cumpla la función**, debe cumplir además otros criterios para ser considerado un buen código JavaScript.

1. ENUNCIADO

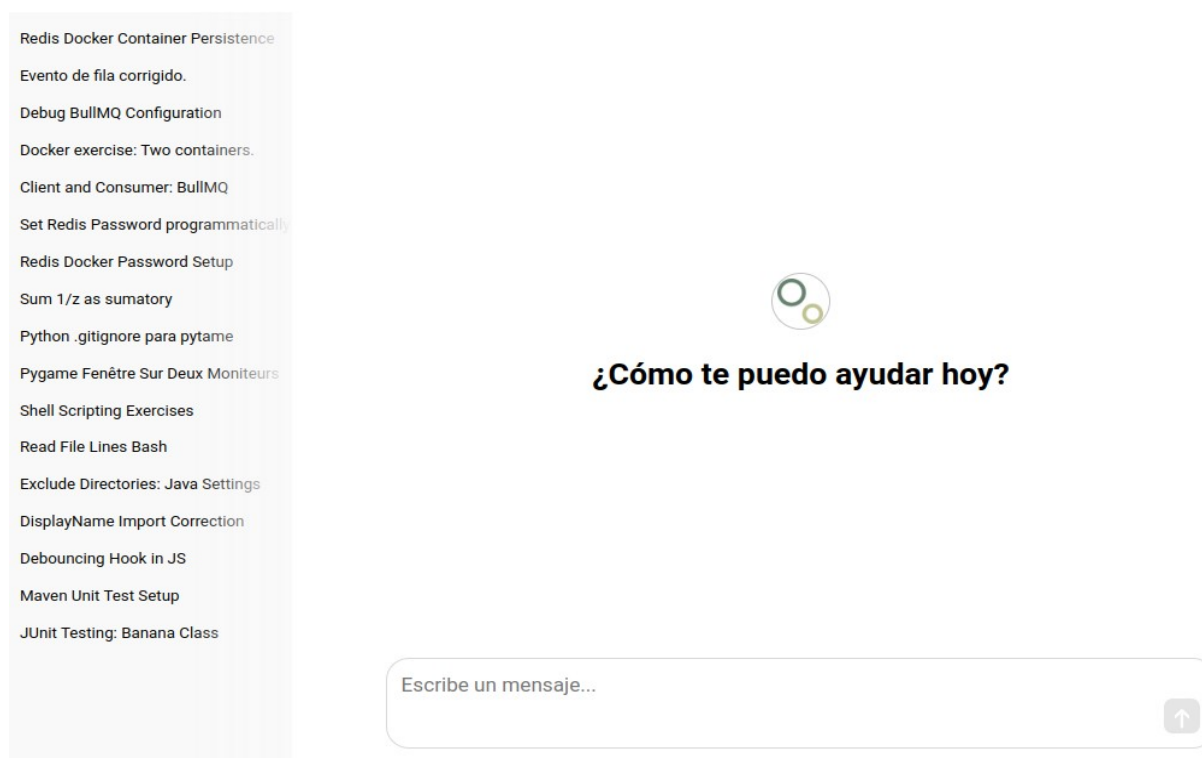
Debes crear una aplicación utilizando React y Redux Toolkit.

Tienes un ejemplo de la aplicación en la siguiente dirección: <https://ceedgpt.netlify.app/>.

El código HTML que debe tener la aplicación, así como el CSS a incluir y los elementos gráficos necesarios están disponibles en el siguiente repositorio:

<https://github.com/CEED-2023/examen-extraordinaria>

La aplicación tendrá el siguiente aspecto:



El funcionamiento será el siguiente:

1.1 Ventana de chat

El botón de la ventana de chat estará con el atributo `disabled` si no hay texto escrito. Cuando se habilite, al pulsarlo se hará lo siguiente:

- Deshabilitar el botón

- Añadir mensaje del usuario en la ventana superior
- Limpiar el texto del campo de entrada
- Enviar una petición al servidor para obtener el mensaje de la IA

Cuando se obtenga el mensaje de la IA se debe añadir la respuesta a la parte superior. Mira el repositorio con el ejemplo de código para ver qué HTML debes añadir para los mensajes, es el mismo para la IA y para el usuario pero cambiando una clase en el `div` de clase `avatar`.

Si la petición de la IA falla debes “devolver” el mensaje del usuario al campo de entrada.

Banner inicial

El banner inicial (“¿Cómo te puedo ayudar hoy?”) sólo se debe mostrar cuando no hay todavía ningún mensaje en la ventana superior.

Redimensión de la ventana de texto

La ventana de texto se ha de redimensionar dinámicamente para ajustarse a la altura del texto escrito. Para ello debes utilizar el hook `useRef`. Tienes una plantilla de cómo utilizarlo en el repositorio del examen en el fichero `TheInputComponent.jsx` (no hace falta que el componente de tu aplicación se llame así) Sólo debes invocar a la función `adjustTextarea` al cargar el componente y cuando el contenido de campo cambie.

1.2 Lista de conversaciones

Cuando se inicie por primera vez la aplicación se cargará la lista de las conversaciones de la izquierda lanzando una llamada al servidor. La lista de conversaciones no se actualizará ni se interactuará con ella, permanecerá todo el tiempo con los datos cargados inicialmente.

2. SERVIDOR DE CHAT

El servidor de chat se encuentra accesible en la siguiente dirección:

<https://examen-extraordinaria-server.fly.dev>

Los endpoints que necesitarás utilizar son:

- **GET** `/chats`: Obtiene la lista de chats.

Respuesta:

```
[
  { "id": 1, "title": "Chat 1" },
  { "id": 2, "title": "Chat 2" },
  ...
]
```

- **POST** `/chat/completions`: Obtiene una respuesta de la IA. Se debe enviar toda la conversación que haya hasta el momento. El servidor responderá con un nuevo mensaje de la IA. El último mensaje de la lista de mensajes enviados ha de ser siempre del usuario. Los roles deben ser `user` para los mensajes del usuario y `assistant` para los de la IA:

Datos de la petición (en formato JSON):

```
{
  "messages": [
    {
      "role": "user",
      "content": "Hola, como estás"
    },
    {
      "role": "assistant",
      "content": "Muy bien, dime en que te puedo ayudar hoy"
    },
    {
      "role": "user",
      "content": "Dime un pareado"
    }
  ]
}
```

Respuesta:

```
{
  "message": "¡Claro! Aquí te dejo uno:\n\n\"El que espera desespera,\ny el que busca encuentra.\"\n\n¿Te gustó?"
}
```

Tienes el código fuente disponible en este repositorio por si necesitas ejecutarlo en local:

<https://github.com/CEED-2023/examen-extraordinaria-server>

3. LINTER

Cada error del linter descontará 3 puntos.

Las reglas del linter son las que están especificadas en `.eslintrc.json` en el repositorio con las plantillas del código. No puede modificarse este fichero. Asimismo no se admitirá deshabilitar ninguna de las reglas de eslint por ningún medio: en ese caso se procederá como si el programa hubiese fallado el linter.

4. ENTREGA

Para la entrega debes eliminar los directorios `node_modules` y `dist` y comprimir el directorio de cada ejercicio en un único fichero `.zip` o `.gz`. Cada ejercicio se entregará en la tarea del curso habilitada a tal efecto.

5. CRITERIOS DE EVALUACIÓN

- El programa es correcto, realiza la función que se solicita en el enunciado
- Se han utilizado estructuras del lenguaje adecuadas: bucles, condicionales, operadores, etc.

- Se han utilizado variables y constantes de forma adecuada
- Se utilizan correctamente y cuando corresponda los tipos de datos y objetos predefinidos del lenguaje (Arrays, objetos planos, Map, Set, etc.)
- Se han utilizado funciones para estructurar el código, definiendo y utilizando parámetros y valores de respuesta de forma adecuada
- El programa es lo más sencillo posible para realizar su función.
- No existe código repetido: se han extraído los comportamientos comunes a funciones y se ha intentado hacer el código genérico.
- La distribución de la aplicación en componentes es adecuada.
- Se ha programado la aplicación utilizando React y Redux toolkit
- El diseño de los stores, actions y reducers es correcto.
- Se utilizan thunks asíncronos apropiadamente.
- El programa cumple todas las reglas definidas para el linter.
- Se genera el HTML solicitado