



UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



Explicativo sobre la Aplicación Educativa de Algoritmos

Fecha:	Noviembre 14 del 2024
Docente:	Alexander Toscano
Presentado por:	Jesús Barrios
	Jorge Mogollón
	Joel Serrano

Explicación sobre como se uso componentes para la maquina tragamonedas

La máquina tragamonedas está construida utilizando componentes en Vue.js, lo que permite que la aplicación sea modular, fácil de mantener y escalable. A continuación, te explicaré cómo se usaron los componentes en la máquina tragamonedas:

Componentización de la Aplicación

Una de las principales ventajas de utilizar Vue.js es la capacidad de dividir la interfaz de usuario en componentes independientes. En la máquina tragamonedas, hemos dividido la interfaz en tres componentes principales:

- SlotScreen.vue: Muestra los carretes que giran.
- LeverAndCredits.vue: Contiene la palanca para iniciar el giro y muestra los créditos restantes.
- PrizeTable.vue: Muestra una tabla con los posibles premios.

**Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.**

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



- Estos componentes están integrados en un componente principal llamado SlotMachine.vue, que los organiza y gestiona la interacción entre ellos.

Ventajas de Usar Componentes

- Modularidad: Cada componente tiene su propio archivo .vue, lo que significa que su estructura (HTML), comportamiento (JavaScript), y estilo (CSS) están encapsulados. Esto facilita el mantenimiento y la actualización de cada parte de la aplicación sin afectar a las demás.
- Reusabilidad: Los componentes pueden ser reutilizados en diferentes partes de la aplicación o incluso en diferentes proyectos. Si en el futuro necesitas una tabla de premios o una pantalla de carretes, puedes reutilizar estos componentes con pequeñas modificaciones.
- Separación de Preocupaciones: Al dividir la aplicación en componentes, cada uno tiene una única responsabilidad. Por ejemplo, SlotScreen.vue se encarga únicamente de manejar la visualización y el comportamiento de los carretes, mientras que LeverAndCredits.vue gestiona los créditos y la palanca.

Cómo se Usaron los Componentes en la Máquina Tragamonedas

a) SlotScreen.vue

Este componente se encarga de mostrar los carretes de la máquina tragamonedas.

- Encapsulación de Estado: El estado de los carretes (reels y displayedReels) está encapsulado dentro del componente. Esto significa que SlotScreen.vue es completamente responsable de cómo se muestran y manejan los carretes.
- Comunicación entre Componentes: El componente escucha un evento (spin-reels) emitido desde LeverAndCredits.vue para iniciar el giro de los carretes. Esto es posible gracias a la emisión de eventos en Vue, que permite la comunicación entre componentes hermanos a través del componente principal.

**Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.**

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co



"VIGILADA MINEDUCACIÓN"

UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



Resolución de MinEduación N° 000020 del 11 de enero de 2023 por 6 años

b) LeverAndCredits.vue

Este componente maneja los créditos y la palanca.

- Manejo de Eventos: Cuando el usuario hace clic en la palanca, se emite un evento (spin-reels) que es escuchado por SlotMachine.vue. Este evento luego es manejado por SlotScreen.vue para iniciar el giro de los carretes.
- Encapsulación de Lógica: La lógica para disminuir los créditos y verificar si el usuario tiene suficientes créditos está encapsulada dentro de este componente, lo que lo hace responsable de todo lo relacionado con los créditos.

c) PrizeTable.vue

Este componente muestra los posibles premios.

- Encapsulación de Datos: Los premios y sus imágenes están encapsulados dentro del componente PrizeTable.vue, lo que significa que si necesitas actualizar los premios o las imágenes, solo necesitas modificar este componente.

Flexibilidad en el Diseño

El uso de componentes permite ajustar fácilmente el diseño y el estilo sin interferir con el comportamiento de los otros componentes. Por ejemplo, si deseas cambiar el diseño de la tabla de premios, puedes hacerlo directamente en PrizeTable.vue sin afectar el resto de la aplicación.

**Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.**

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co



"VIGILADA MINEDUCACIÓN"

UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



Resolución de MinEduación N° 000020 del 11 de enero de 2023 por 6 años

Integración en SlotMachine.vue

El componente SlotMachine.vue actúa como el contenedor principal que organiza y coordina los otros componentes.

Integración de Componentes

- SlotMachine.vue importa y utiliza SlotScreen.vue, LeverAndCredits.vue, y PrizeTable.vue dentro de su template. Esto crea la estructura visual de la máquina tragamonedas.

Comunicación entre Componentes

- SlotMachine.vue actúa como intermediario en la comunicación entre LeverAndCredits.vue y SlotScreen.vue. Recibe el evento spin-reels de LeverAndCredits.vue y lo maneja llamando al método spinReels() en SlotScreen.vue.

Página Principal (index.vue)

Finalmente, la página index.vue es donde se integra el componente SlotMachine.vue.

Encapsulación de la Lógica Principal

La página index.vue simplemente incluye el componente SlotMachine.vue, encapsulando así toda la lógica y presentación de la máquina tragamonedas en un solo componente.

**Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.**

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co



"VIGILADA MINEDUCACIÓN"

UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



Conclusión

El uso de componentes en Vue.js ha permitido que la máquina tragamonedas esté organizada en partes pequeñas y manejables. Cada componente se encarga de una parte específica de la aplicación, lo que facilita el desarrollo, la depuración y el mantenimiento. Además, la comunicación entre componentes a través de eventos hace que la interacción sea fluida y modular, permitiendo la separación clara de responsabilidades y la posibilidad de reutilizar código en el futuro.

Proyecto de Visualización de Respuestas en JSON

Este proyecto permite mostrar una lista de respuestas con información de su nivel de calidad. Usa Nuxt 3 como framework principal, Pinia para el manejo del estado y una API simulada para la carga de datos en formato JSON.

Estructura del Proyecto

Components/Respuestas.vue: Componente de visualización de respuestas.

Server/api/consultas/index.get.ts: Archivo de la API que devuelve datos de respuestas en JSON.

stores/respuestas.ts: Store de Pinia que maneja el estado y la obtención de respuestas.

Pages/index.vue: Página principal que integra el componente Respuestas.

*Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.*

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co



"VIGILADA MINEDUCACIÓN"

UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



La siguiente imagen representa la página principal de la aplicación y utiliza el componente Respuestas para mostrar las respuestas.

```
pages > index.vue > { } script setup
1  <template>
2    <h1>Principal</h1>
3    <Respuestas />
4  </template>
5
6  <script setup>
7    // Importa el componente `Respuestas.vue`
8    import Respuestas from '~/components/respuestas.vue'
9  </script>
10
```

Explicación de la Página Principal

Uso de <Respuestas />:

Esta página incluye el componente Respuestas, que se encarga de mostrar la lista de respuestas. El componente Respuestas se importa y se inserta en el template para que se muestre en la página.

**Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.**

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co



"VIGILADA MINEDUCACIÓN"

UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



Función

La página index.vue sirve como punto de entrada a la aplicación, donde se visualizan las respuestas cargadas. Este componente es responsable de mostrar la lista de respuestas obtenidas del store respuestas.

```
components > respuestas.vue > {} script setup
1  <template>
2    <div>
3      <h1>Respuestas</h1>
4      <div v-if="getRespuestas.length">
5        <ul>
6          <li v-for="respuesta in getRespuestas" :key="respuesta.id">
7            <h2>{{ respuesta.title }}</h2>
8            <p>{{ respuesta.description }}</p>
9            <strong>Nivel: {{ respuesta.level }}</strong>
10           </li>
11         </ul>
12       </div>
13       <div v-else>
14         <p>No hay respuestas disponibles.</p>
15       </div>
16     </div>
17 </template>
18
19 <script setup>
20   // Importa el store que configuraste
21   import { useRespuestasStore } from "~/stores/respuestas";
22
23   // Instancia el store
24   const respuestasStore = useRespuestasStore();
25   const { getRespuestas } = respuestasStore;
26
27   // Llama a `fetchRespuestas` para cargar los datos
28   await respuestasStore.fetchRespuestas();
29 </script>
30
```

*Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.*

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co



UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



Resolución de MinEduación N° 000020 del 11 de enero de 2023 por 6 años

Explicación del Componente

<template>:

Muestra un encabezado "Respuestas" para indicar el propósito del componente. Si hay respuestas disponibles (`getRespuestas.length > 0`), recorre la lista y muestra cada respuesta con su título (`title`), descripción (`description`) y nivel de calidad (`level`). Si no hay respuestas, muestra un mensaje indicando que no hay datos disponibles.

Script

Store: Importa y usa el store `useRespuestasStore` para acceder a las respuestas.

Datos: Llama a la acción `fetchRespuestas` para cargar los datos desde la API al montarse el componente.

getRespuestas: Es un getter del store que permite acceder a la lista de respuestas cargadas.

Este componente está diseñado para ser reutilizable y se puede incorporar en cualquier página que requiera mostrar las respuestas.

Server/api/consultas/index.get.ts

Este archivo contiene la lógica del endpoint de la API. Se trata de un archivo de servidor en la carpeta `server` y se encarga de devolver un JSON con datos simulados de respuestas.

```
server > api > consultas > TS index.get.ts > ...
1  export default defineEventHandler(async (event) => {
2    const respuestas = [
3      { id: 1, title: 'Respuesta 1', level: 'Muy buena', description: 'Descripción de la respuesta 1' },
4      { id: 2, title: 'Respuesta 2', level: 'Buena', description: 'Descripción de la respuesta 2' },
5      { id: 3, title: 'Respuesta 3', level: 'Mala', description: 'Descripción de la respuesta 3' }
6    ];
7
8    return respuestas;
9  });
10
```

*Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.*

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co



"VIGILADA MINEDUCACIÓN"

UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



Explicación del Endpoint

Propósito: Este archivo simula una API que devuelve una lista de respuestas en formato JSON. Cada respuesta incluye:

id: Identificador único de cada respuesta.

title: Título de la respuesta.

level: Indica el nivel o calidad de la respuesta (por ejemplo, "Muy buena", "Buena", "Mala").

description: Breve descripción de la respuesta.

Funcionamiento

Al acceder a /api/consultas, este endpoint devuelve el array de respuestas. El método `defineEventHandler` permite manejar las solicitudes a este endpoint y devolver la lista de respuestas simulada. Este archivo facilita el desarrollo sin necesidad de una base de datos real, ya que simula los datos.

¿Cómo funciona una API?



*Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.*

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co



"VIGILADA MINEDUCACIÓN"

UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



Stores/respuestas.ts

Este archivo define el store use Respuestas Store utilizando Pinia para manejar el estado de las respuestas de manera centralizada.

```
stores > TS respuestas.ts > useRespuestasStore > actions > fetchRespuestas > respuestas
1  import { defineStore } from "pinia";
2
3  interface Respuesta {
4    id: number;
5    title: string;
6    level: string;
7    description: string;
8  }
9
10 export const useRespuestasStore = defineStore("respuestas", {
11   state: () => ({
12     respuestas: [] as Respuesta[], // Define `respuestas` como un array de tipo Respuesta
13   }),
14   getters: {
15     getRespuestas: (state) => state.respuestas,
16   },
17   actions: {
18     async fetchRespuestas() {
19       try {
20         const respuestas = await $fetch("/api/respuestas");
21         this.respuestas = respuestas;
22       } catch (error) {
23         console.error("Error al obtener las respuestas:", error);
24       }
25     },
26   },
27 });
28
```

*Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.*

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co



"VIGILADA MINEDUCACIÓN"

UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



Resolución de MinEduación N° 000020 del 11 de enero de 2023 por 6 años

Explicación del Store

Interfaz Respuesta: Define la estructura de los datos de una respuesta, asegurando que cada respuesta tenga id, title, level y description.

Estado (state)

respuestas: Array que almacena la lista de respuestas obtenida de la API.

Getters

getRespuestas: Devuelve la lista de respuestas del estado, permitiendo el acceso al array respuestas.

Acciones

fetchRespuestas: Acción que obtiene datos de /api/consultas usando \$fetch.

Control de Errores: Muestra un mensaje en consola si ocurre algún error al intentar obtener los datos.

El store permite la gestión centralizada de las respuestas y asegura que se puedan reutilizar en múltiples componentes si es necesario.

Resumen para la Entrega

Propósito General: Esta aplicación permite cargar, almacenar y visualizar una lista de respuestas con su nivel de calidad usando Nuxt 3, Pinia y una API simulada.

*Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.*

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co



"VIGILADA MINEDUCACIÓN"

UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



Resolución de MinEduación N° 000020 del 11 de enero de 2023 por 6 años

Componentes Clave

- **Respuestas.vue:** Muestra la lista de respuestas en la página principal.
- **index.get.ts (API):** Proporciona una lista de respuestas en formato JSON desde un endpoint local.
- **respuestas.ts (Store):** Centraliza el estado de respuestas y facilita su acceso desde distintos componentes.
- **index.vue:** Página principal que carga y muestra el componente Respuestas.

Este documento te permitirá explicar el funcionamiento de la aplicación y sus componentes en detalle. Con esta estructura, tienes una base sólida para construir aplicaciones de datos en Nuxt y Pinia.

Link del repositorio

<https://github.com/misterx1999/Proyecto-de-Visualizaci-n-de-Respuestas-en-JSON.git>

*Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.*

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co



"VIGILADA MINEDUCACIÓN"

UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



Resolución de MinEduación N° 000020 del 11 de enero de 2023 por 6 años

Documentación de la Interacción con la API en Edupro

*Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.*

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co

Componentes Relacionados con la API

DocenteRespuestas.vue

Este componente se utiliza para que los docentes puedan visualizar las respuestas enviadas por los estudiantes y calificarlas.

Código de DocentesRespuestas.vue

```
components > ▼ DocenteRespuestas.vue > ...
1  <template>
2    <v-container>
3      <h1>Respuestas de Estudiantes</h1>
4      <v-list>
5        <v-list-item v-for="respuesta in respuestas" :key="respuesta.id">
6          <v-list-item-content>
7            <v-list-item-title>{{ respuesta.studentName }}</v-list-item-title>
8            <v-list-item-subtitle>{{ respuesta.description }}</v-list-item-subtitle>
9          </v-list-item-content>
10         <v-list-item-action>
11           <v-select
12             v-model="respuesta.valoracion"
13             :items="['Muy buena', 'Buena', 'Regular', 'Mala']"
14             label="Valorar"
15             @change="valorarRespuesta(respuesta.id, respuesta.valoracion)"
16           />
17         </v-list-item-action>
18       </v-list-item>
19     </v-list>
20   </v-container>
21 </template>
22
23 <script setup>
24   import { useRespuestasStore } from "@stores/respuestas";
25
26   const respuestasStore = useRespuestasStore();
27   const respuestas = computed(() => respuestasStore.respuestas);
28
29   const valorarRespuesta = async (id, valoracion) => {
30     await respuestasStore.valorarRespuesta(id, valoracion);
31   };
32 </script>
33 |
```

Funcionalidad

Obtiene las respuestas de la API mediante el store respuestas.ts.
Permite a los docentes valorar respuestas enviadas por los estudiantes.

Pages Relacionadas con la API



UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



docentes.vue

Esta página utiliza el componente DocenteRespuestas.vue para mostrar las respuestas y permitir la valoración.

Código de Docentes.vue

```
pages > ▼ docentes.vue > ...
1  <template>
2    <DocenteRespuestas />
3  </template>
4
5  <script setup>
6    import DocenteRespuestas from "~/components/DocenteRespuestas.vue";
7
8    definePageMeta({
9      middleware: "auth",
10     allowedRoles: ["docente"],
11   });
12 </script>
13
```

Funcionalidad

- Redirige automáticamente a esta página si el usuario tiene el rol de docente.
- Carga el componente **DocenteRespuestas** para mostrar las respuestas.

algoritmos.vue

Esta página permite a los estudiantes enviar sus respuestas, las cuales se almacenan en la API.

*Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.*

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co



"VIGILADA MINEDUCACIÓN"

UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



```
pages > algoritmos.vue > ...
1 <template>
2   <v-container class="algoritmos-background">
3     <h2>Ordena el Algoritmo Correctamente</h2>
4     <p>Arrastra y suelta los pasos del algoritmo en el orden correcto:</p>
5
6     <v-row class="my-4">
7       <v-col v-for="(step, index) in shuffledSteps" :key="index">
8         <AlgoritmoStep :stepText="step.text" :index="index" @swapSteps="swapSteps" />
9       </v-col>
10    </v-row>
11
12    <AlgoritmoChecker :steps="shuffledSteps" :correctOrder="correctOrder" />
13
14    <VideoControl />
15
16    <ActividadAlgoritmo @completedActivity="handleActivityCompletion" />
17
18    <v-expansion-panels v-if="showExpansionPanels">
19      <v-expansion-panel title="¿Qué es un algoritmo?">
20        <v-expansion-panel-text>
21          Un algoritmo es una secuencia de pasos diseñada para resolver un problema.
22        </v-expansion-panel-text>
23      </v-expansion-panel>
24    </v-expansion-panels>
25  </v-container>
26 </template>
27
28 <script setup>
29   import AlgoritmoStep from "@components/AlgoritmoStep.vue";
30   import AlgoritmoChecker from "@components/AlgoritmoChecker.vue";
31   import VideoControl from "@components/VideoControl.vue";
32   import ActividadAlgoritmo from "@components/ActividadAlgoritmo.vue";
33   import { ref, onMounted } from "vue";
34
35   const shuffledSteps = ref([]);
36   const correctOrder = ref([]);
37   const showExpansionPanels = ref(false);
38
39   onMounted(async () => {
40     const response = await fetch("/algoritmos.json");
41     const data = await response.json();
42
43     if (data[0]?.steps) {
44       correctOrder.value = data[0].steps;
45       shuffledSteps.value = [...data[0].steps].sort(() => Math.random() - 0.5);
46     }
47   });
48
49   const swapSteps = ({ fromIndex, toIndex }) => {
50     const temp = shuffledSteps.value[fromIndex];
51     shuffledSteps.value[fromIndex] = shuffledSteps.value[toIndex];
52     shuffledSteps.value[toIndex] = temp;
53   };
54
55   const handleActivityCompletion = () => {
56     showExpansionPanels.value = true;
57   };
58
59   definePageMeta({
60     middleware: "auth",
61   });
62 </script>
```

Funcionalidad

- Los estudiantes envían sus respuestas desde el componente

*Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.*

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co

ActividadAlgoritmo.vue.

- Las respuestas se envían al store y posteriormente a la API.

Stores Relacionados con la API

respuestas.ts

- Este archivo maneja la lógica de almacenamiento, obtención y valoración de las respuestas.

```
stores > TS respuestas.ts > [x] useRespuestasStore > [x] defineStore('respuestas') callback
1  import { defineStore } from 'pinia';
2  import { ref } from 'vue';
3
4  // Define el tipo de una respuesta
5  interface Respuesta {
6    id: number;
7    title: string;
8    level: string;
9    description: string;
10   valoracion: string | null;
11 }
12
13 // Define el store
14 export const useRespuestasStore = defineStore('respuestas', () => {
15   const respuestas = ref<Respuesta[]>([]);
16
17   const fetchRespuestas = async () => {
18     const data: Respuesta[] = await $fetch('/api/respuestas');
19     respuestas.value = data;
20   };
21
22   const addRespuesta = async (newRespuesta: Respuesta) => {
23     respuestas.value.push(newRespuesta);
24   };
25
26   const valorarRespuesta = async (id: number, valoracion: string) => {
27     respuestas.value = respuestas.value.map((respuesta) => {
28       respuesta.id === id ? { ...respuesta, valoracion } : respuesta
29     });
30   };
31
32   return { respuestas, fetchRespuestas, addRespuesta, valorarRespuesta };
33 });
```

Funcionalidad

- **fetchRespuestas:** Obtiene respuestas desde la API.
- **addRespuesta:** Agrega una nueva respuesta enviada por un estudiante.
- **valorarRespuesta:** Permite valorar las respuestas obtenidas.

Servidor (Server)

index.get.ts

- Este archivo devuelve un array de respuestas simuladas desde la API.

```
server > api > respuestas > ts index.get.ts > ...
1  export default defineEventHandler(async () => {
2      const respuestas = [
3          { id: 1, title: 'Respuesta 1', level: 'Muy buena', description: 'Descripción de la respuesta 1', valoracion: null },
4          { id: 2, title: 'Respuesta 2', level: 'Buena', description: 'Descripción de la respuesta 2', valoracion: null },
5          { id: 3, title: 'Respuesta 3', level: 'Mala', description: 'Descripción de la respuesta 3', valoracion: null }
6      ];
7
8      return respuestas;
9  });
10
```

Funcionalidad

- Proporciona un endpoint para obtener respuestas simuladas.

index.post.ts

- Este archivo simula la actualización o adición de respuestas en la



"VIGILADA MINEDUCACIÓN"

UNIVERSIDAD DE CÓRDOBA

LICENCIATURA EN INFORMÁTICA Y MEDIOS
AUDIOVISUALES

TÉCNICAS AVANZADAS DE PROGRAMACIÓN



Resolución de MinEduación N° 000020 del 11 de enero de 2023 por 6 años

API.

```
server > api > respuestas > TS index.postts > ...  
1  export default defineEventHandler(async (event) => {  
2    const body = await readBody(event);  
3    // Simular actualización (en producción se conectaría a una base de datos)  
4    return {  
5      message: 'Respuesta actualizada con éxito',  
6      updated: body,  
7    };  
8  });  
9
```

Funcionalidad

- Permite simular la actualización de una respuesta.
- Utiliza el cuerpo de la solicitud para actualizar los datos.

*Unicórdoba, calidad, innovación e inclusión
para la transformación del territorio.*

Carrera 6ª. No. 77-30 Montería - NIT. 891080031-3
www.unicordoba.edu.co