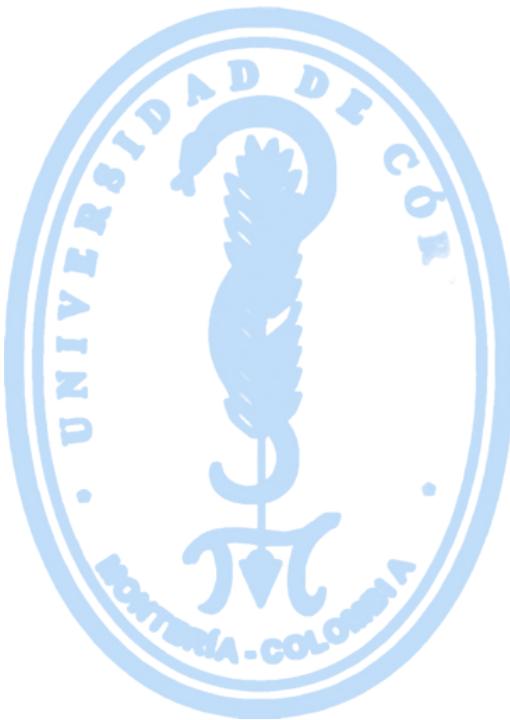


LICENCIATURA EN
INFORMÁTICA

Acreditada de Alta Calidad
MEN Res. 10710 25/05/17



INFORME FINAL

EMOTIEDU

SELENIS ACOSTA PATERNIAN
MARIA ALEJANDRA SOTO SANCHEZ
MARINELLY PÉREZ SALCEDO

**TECNICAS AVANZADAS DE
PROGRAMACIÓN**

ALEXANDER TOSCANO RICARDO

**UNIVERSIDAD DE CÓRDOBA
SEMESTRE V
2024**



INTRODUCCIÓN

El presente informe sustenta el desarrollo de la aplicación "**EmotiEdu**" una plataforma centrada en el bienestar de sus respectivos usuarios y la gestión de sus emociones. Su principal propósito es fortalecer la inteligencia emocional, y concientizar a las personas de la importancia de educarse en salud mental. Este espacio está diseñado para fomentar el autoconocimiento. A través de una serie de recursos multimedia, el objetivo es mejorar el bienestar mental de manera integral, facilitando el acceso a herramientas para el manejo de emociones en diferentes situaciones.

DATOS RELEVANTES

- Un informe de UNESCO indica que menos del 25% de los sistemas educativos del mundo incluye educación sobre salud mental en sus currículos.
- Según la OMS, 76-85% de las personas en países de ingresos bajos y medios con trastornos mentales graves no reciben tratamiento adecuado, en parte debido a la falta de educación pública sobre salud mental.
- Un estudio de 2023 por The Lancet reveló que el 64% del contenido sobre salud mental en redes sociales incluye información errónea, lo que dificulta que las personas encuentren recursos confiables.

OBJETIVOS

Esta aplicación es creada con el objetivo de proporcionar un espacio accesible para todo público donde puedan fortalecer su inteligencia emocional a través de recursos interactivos, la página busca fomentar el autoconocimiento emocional, el manejo saludable de las emociones y la resiliencia en momentos de dificultad, promoviendo el bienestar mental y emocional de manera integral.



DESCRIPCIÓN DE LAS PÁGINAS

Inicio

En la página de Inicio, se presenta la vista principal de la aplicación. Esta incluye el nombre de la plataforma, una frase inspiradora y una imagen representativa que enfatiza la importancia de cuidar la nutrición de nuestra mente.

The screenshot shows the EmotiEdu homepage. At the top, there's a purple header bar with the text "≡ VISTAS". Below it is the main content area. On the left, the EmotiEdu logo is displayed in large purple and blue letters. Below the logo is a quote in bold black text: "La verdadera fortaleza no está en no caer, sino en levantarse cada vez que lo haces". Underneath the quote is a blue button labeled "INICIAR SESIÓN". On the right side of the main area, there's a large, stylized illustration of a human head in profile, facing right. The head is filled with various colorful flowers and plants, symbolizing mental well-being. A man in a blue shirt and orange pants is standing on a ladder, watering the flowers from a red watering can. A woman in an orange top and blue skirt is walking towards the right side of the head. The background of the main area is white, and the overall aesthetic is clean and modern.

© 2024 EmotiEdu. Todos los derechos reservados

Exploración Emocional

En esta sección, el usuario podrá seleccionar la emoción que predomina en su estado actual para acceder a contenidos personalizados. Se presentará una lista de emociones. Al elegir una emoción, se desplegarán recursos recomendados diseñados específicamente para ayudar al usuario a gestionar y comprender mejor ese estado emocional.

The screenshot shows the Emotional Exploration section. At the top, there's a purple header bar with the text "≡ VISTAS". Below it is the main content area. In the center, the text "¿CÓMO TE SIENTES HOY?" is displayed in pink and blue. Below this text is a grid of twelve emoji-like icons representing different emotions, arranged in two rows of six. The first row includes icons for happiness, sadness, anger, and others. The second row includes icons for fear, disgust, and other expressions. Below the grid, there's a large, semi-transparent overlay card. The card has a purple header with the word "Enojado" (Angry). It features a cartoon illustration of a red, angry face with a wide-open mouth and bulging eyes. To the left of the face is a small logo for "Diccionario de las Emociones". To the right, there's a rainbow and the word "Compartir" (Share). At the bottom of the card, there's a play button icon, suggesting a video or interactive content. The footer of the card contains the text "Diccionario de las Emociones: Enojo" and "Compartir". At the very bottom of the entire screenshot, there's a blue footer bar with the text "© 2024 EmotiEdu. Todos los derechos reservados".



DESCRIPCIÓN DE LAS PÁGINAS

Centro de Bienestar

En la página de Inicio, se presenta la vista principal de la aplicación. Esta incluye el nombre de la plataforma, una frase inspiradora y una imagen representativa que enfatiza la importancia de cuidar la nutrición de nuestra mente.

The screenshot shows the main interface of the EmotiEdu app. At the top, there is a purple header bar with the word "VISTAS". Below it, there are two video thumbnails. The first video is titled "¿Por qué nos equivocamos en el amor?" and the second is titled "Relaciones", both featuring a man sitting on a chair. Below these are three Spotify episode cards: "Conflictos y Heridas Emocionales", "Poner Límites", and "Acompañamiento Emocional", each with a play button and a "Guardar en Spotify" option. At the bottom of the screen, there is a blue footer bar with the text "© 2024 EmotiEdu. Todos los derechos reservados".

Área de Emergencia

En esta sección, encontrarás acceso rápido a recursos de apoyo inmediato, incluyendo enlaces a líneas telefónicas y chats diseñados para brindar asistencia en casos de emergencia emocional.

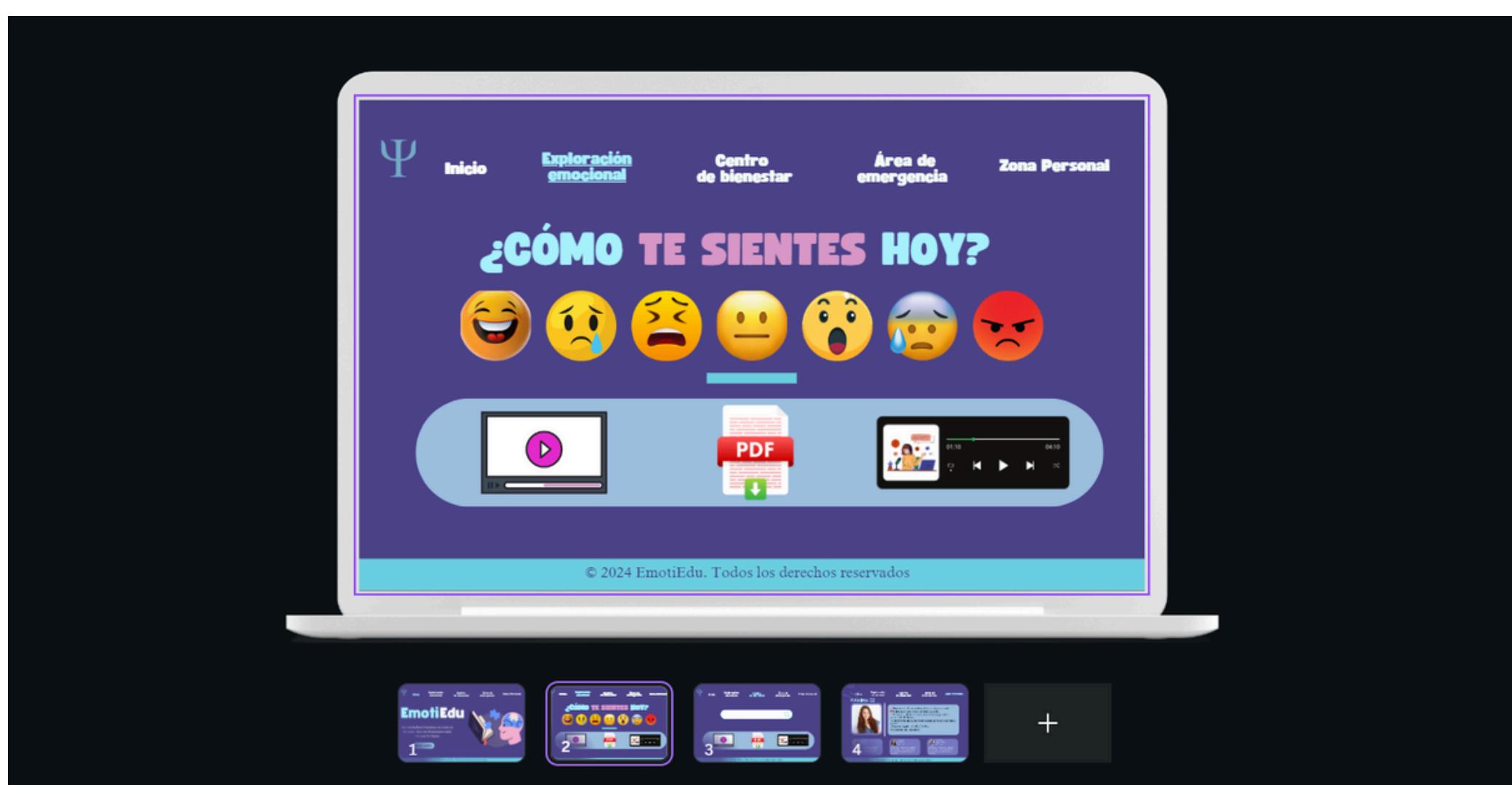
The screenshot shows the "Zona de Emergencia Emocional" section. It features a large white box labeled "Zona de Emergencia Emocional" with the sub-instruction "Aquí puedes encontrar apoyo inmediato en situaciones críticas.". Below this are two smaller boxes: "Recursos de Ayuda Inmediata" and "Foro de Apoyo". The "Recursos de Ayuda Inmediata" box contains links to "Recursos para gestionar ansiedad, tristeza, etc." (with a "IR" button) and "Técnicas para relajarte." (with a "IR" button). The "Foro de Apoyo" box has a text input field for users to share their experiences, a green "ENVIAR" button, and a link to "Cómo dominar tu mente: el sencillo entrenamiento..." with a "IR" button. At the bottom, there are two banners: one for "wake up" with the tagline "Cómo tener paz mental, elimina la preocupación ..." and another for "Cómo dominar tu mente: el sencillo entrenamiento...".



Barra de Navegación



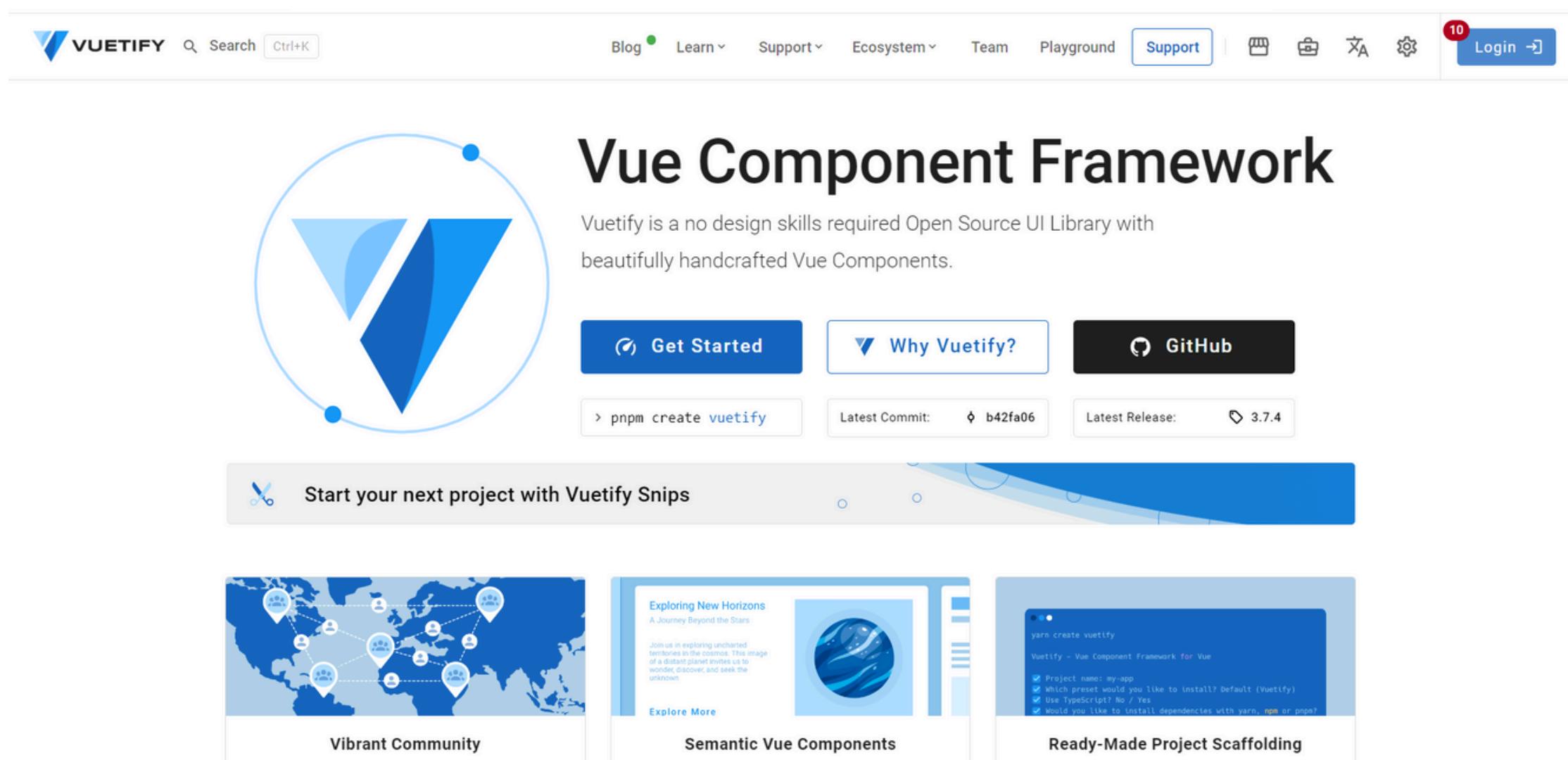
DISEÑO DE MOCKUP



LINK:

https://www.canva.com/design/DAGP-3aiME4/K35ZJICyaG5li7K1esF7aA/edit?utm_content=DAGP3aiME4&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

IMPLEMENTACIÓN DE VUETIFY



El desarrollo de la aplicación utilizando Vuetify comienza con la selección de los componentes preexistentes que mejor se ajusten a las funcionalidades especificadas en el mockup. Esto incluye la identificación de las interfaces necesarias, lo que implica definir las páginas o pantallas que formarán parte de la aplicación. A partir de esta información, se realiza un análisis detallado para decidir qué componentes serán imprescindibles en cada página, considerando tanto los componentes básicos como los adicionales. Estos pueden incluir botones, formularios, barras de navegación, entre otros.

Una vez identificados los componentes, el siguiente paso es integrarlos de manera básica en la aplicación para evaluar su funcionalidad. Esta integración permite verificar si los componentes seleccionados cumplen con las expectativas de diseño y funcionalidad, lo que asegura que se alinean con los requerimientos iniciales. Después de la integración inicial, se procede a la personalización y adaptación de los componentes. Este proceso implica modificar los estilos, comportamientos y características de los componentes de Vuetify para ajustarlos de forma precisa a las necesidades y especificaciones del proyecto, garantizando que la aplicación no solo sea funcional, sino también visualmente coherente con el diseño propuesto.



CARPETAS Y ARCHIVOS DEL PROYECTO

Components

```
└─ components
    └─ Bienes
        └─ Audios.vue
        └─ Buscar.vue
        └─ Videos.vue
    └─ Emergencia
        └─ contactos.vue
        └─ recursos.vue
        └─ titulo.vue
    └─ Explo
        └─ Emociones
            └─ EmojiButton.vue
    └─ Inicio
        └─ Bienvenida
            └─ Imagen.vue
            └─ Texto.vue
            └─ Button.vue
            └─ Pie.vue
```

Pages

```
└─ pages
    └─ Bienestar.vue
    └─ Emergencia.vue
    └─ Exploracion.vue
    └─ index.vue
    └─ Personal.vue
    └─ Products.vue
```

Server/Api

```
└─ server
    └─ api
        └─ bienestar
            └─ audio.get.ts
            └─ index.get.ts
        └─ emotions
            └─ index.get.ts
        └─ products
            └─ tsconfig.json
```

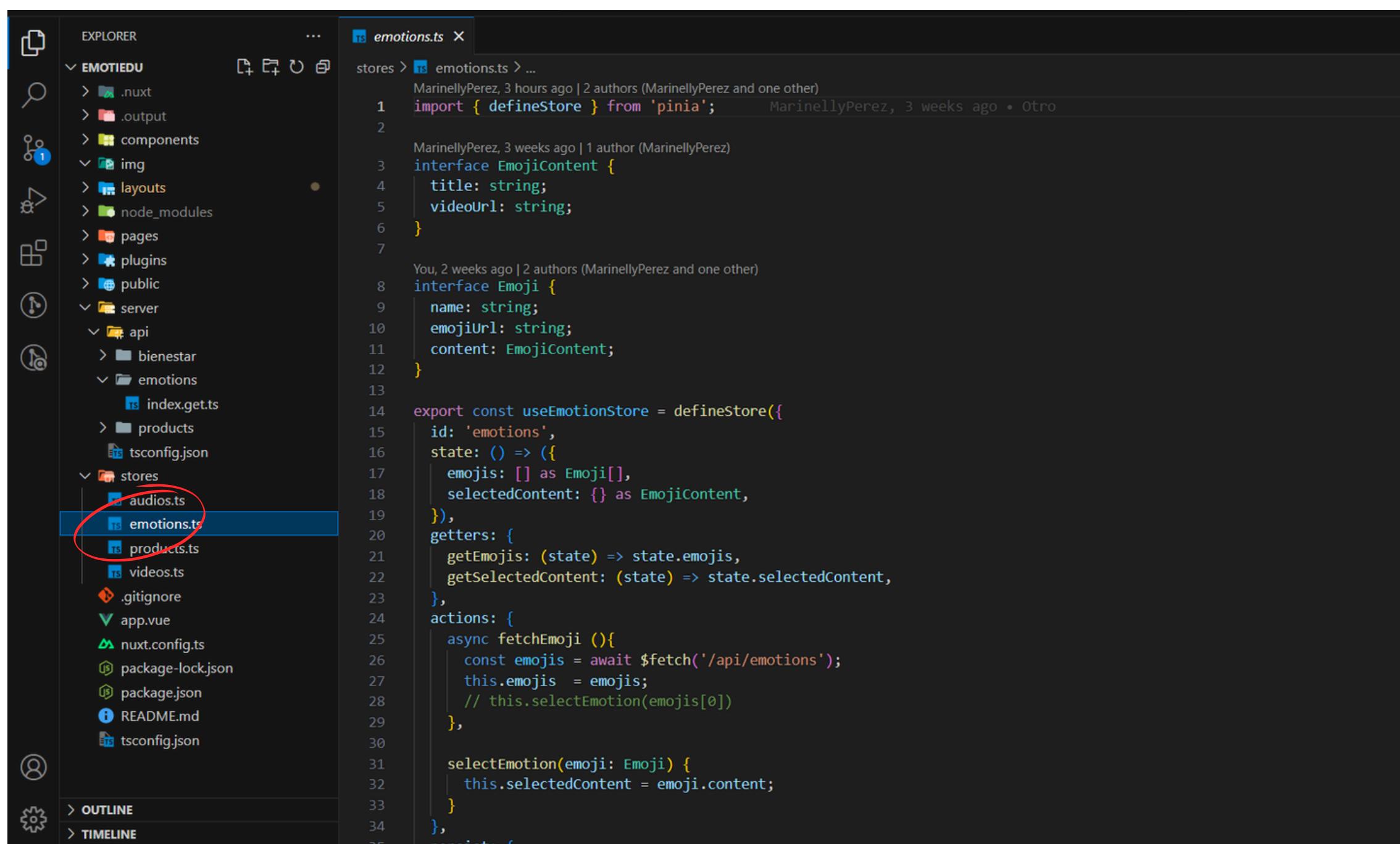
Stores

```
└─ stores
    └─ audios.ts
    └─ emotions.ts
    └─ products.ts
    └─ videos.ts
```



IMPLEMENTACIÓN DEL STATE

Un state (estado) se refiere a la representación de los datos que definen la situación o condición de una aplicación o componente en un momento dado. El state almacena información sobre la cual se pueden realizar operaciones, y puede ser modificado en respuesta a las interacciones del usuario, cambios internos o eventos.



```
stores > emotions.ts > ...
MarinellyPerez, 3 hours ago | 2 authors (MarinellyPerez and one other)
1 import { defineStore } from 'pinia';
2
MarinellyPerez, 3 weeks ago | 1 author (MarinellyPerez)
3 interface EmojiContent {
4   title: string;
5   videoUrl: string;
6 }
7
You, 2 weeks ago | 2 authors (MarinellyPerez and one other)
8 interface Emoji {
9   name: string;
10  emojiUrl: string;
11  content: EmojiContent;
12 }
13
14 export const useEmotionStore = defineStore({
15   id: 'emotions',
16   state: () => ({
17     emojis: [] as Emoji[],
18     selectedContent: {} as EmojiContent,
19   }),
20   getters: {
21     getEmojis: (state) => state.emojis,
22     getSelectedContent: (state) => state.selectedContent,
23   },
24   actions: {
25     async fetchEmoji () {
26       const emojis = await $fetch('/api/emotions');
27       this.emojis = emojis;
28       // this.selectEmotion(emojis[0])
29     },
30
31     selectEmotion(emoji: Emoji) {
32       this.selectedContent = emoji.content;
33     }
34   }
});
```

CREACIÓN DE LAS API

Un API (Interfaz de Programación de Aplicaciones, por sus siglas en inglés) es un conjunto de definiciones y protocolos que permiten que diferentes aplicaciones o sistemas se comuniquen entre sí. En otras palabras, un API es una forma estandarizada de que un software acceda a funcionalidades o datos de otro software, sin necesidad de conocer su implementación interna.

- Se creó una Api para cada uno de los componentes que integran las páginas

The screenshot shows the Visual Studio Code interface with the following details:

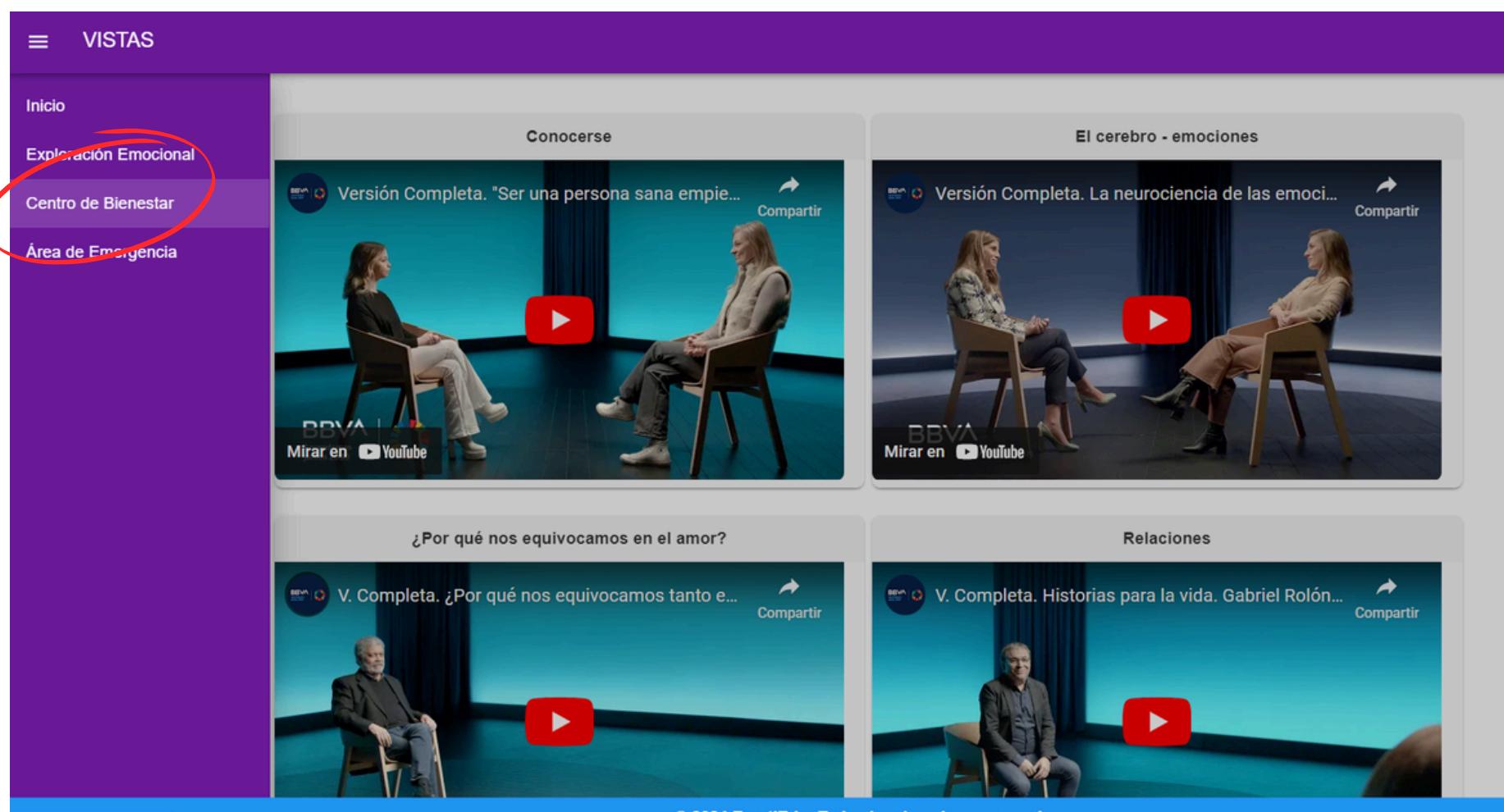
- Explorer View:** On the left, the file tree shows the project structure. A red oval highlights the `server` folder, which contains `api`, `bienestar`, and `emotions`. The `index.get.ts` file is selected within the `emotions` folder.
- Code Editor:** The main area displays the `index.get.ts` file content. The code defines an `index.get.ts` function that returns an array of emotion objects. Each emotion object has a name, emoji URL, and content (title and video URL). The code uses `<iframe>` tags to embed YouTube videos.

```
server > api > emotions > index.get.ts > default > defineEventHandler() callback
MarinellyPerez, 3 hours ago | 1 author (MarinellyPerez)
export default defineEventHandler(async (event)=>[
  const emotions = [
    {
      name: "Euforia",
      emojiUrl: "https://imagedelivery.net/zMsc4Rh1AHFp0qz_tfBn9w/png_08--nhQBoAZrZbbQ-QCUL/public?id=2024-12-03T02:49:48.630Z",
      content: {
        title: "Eufórico",
        videoUrl: '<iframe width="560" height="315" src="https://www.youtube.com/embed/2wzXgRbxqVs?si=sfwWjAqaIYCrkrFwi" title="YouTube video player"></iframe>',
      }
    },
    {
      name: "Alegria",
      emojiUrl: "https://imagedelivery.net/zMsc4Rh1AHFp0qz_tfBn9w/png_01--NG0gi4uktQPkshthu/public?id=2024-12-03T02:55:54.155Z",
      content: {
        title: "Alegre",
        videoUrl: '<iframe width="560" height="315" src="https://www.youtube.com/embed/hyetfmz3gcM?si=M5efmaCxvRf6MqGf" title="YouTube video player"></iframe>',
      }
    },
    {
      name: "Confianza",
      emojiUrl: "https://imagedelivery.net/zMsc4Rh1AHFp0qz_tfBn9w/png_17--t0bmeV6sw7hDZukQD/public?id=2024-12-03T03:22:35.255Z",
      content: {
        title: "Seguro",
        videoUrl: '<iframe width="560" height="315" src="https://www.youtube.com/embed/gJOJKStkBYU?si=WhY8wwVp3G_T7Vp9" title="YouTube video player"></iframe>',
      }
    },
    {
      name: "Aburrimiento",
      emojiUrl: "https://imagedelivery.net/zMsc4Rh1AHFp0qz_tfBn9w/png_40--eQpSY7nwgH0vzsWBU/public?id=2024-12-03T03:00:53.392Z",
      content: {
        title: "Aburrido",
        videoUrl: '<iframe width="560" height="315" src="https://www.youtube.com/embed/b9u_1XoNgJo?si=-dnJrKxIyNETl2ux" title="YouTube video player"></iframe>',
      }
    },
    {
      name: "Indiferencia",
      emojiUrl: "https://imagedelivery.net/zMsc4Rh1AHFp0qz_tfBn9w/png_39--eQpSY7nwgH0vzsWBU/public?id=2024-12-03T03:00:53.392Z",
      content: {
        title: "Indiferente",
        videoUrl: '<iframe width="560" height="315" src="https://www.youtube.com/embed/2wzXgRbxqVs?si=sfwWjAqaIYCrkrFwi" title="YouTube video player"></iframe>',
      }
    }
  ]
]
```



RELACIÓN ENTRE EL FLUJO DE DATOS

1. Solicitar datos a través de una API: En una aplicación, cuando un usuario interactúa con la interfaz (por ejemplo, haciendo clic en un botón o cargando una página), puede haber una necesidad de obtener datos de un servidor o servicio externo.



Para esto, la aplicación hace una solicitud a la API.

2. El API responde con datos: La API devuelve los datos solicitados en un formato como JSON.

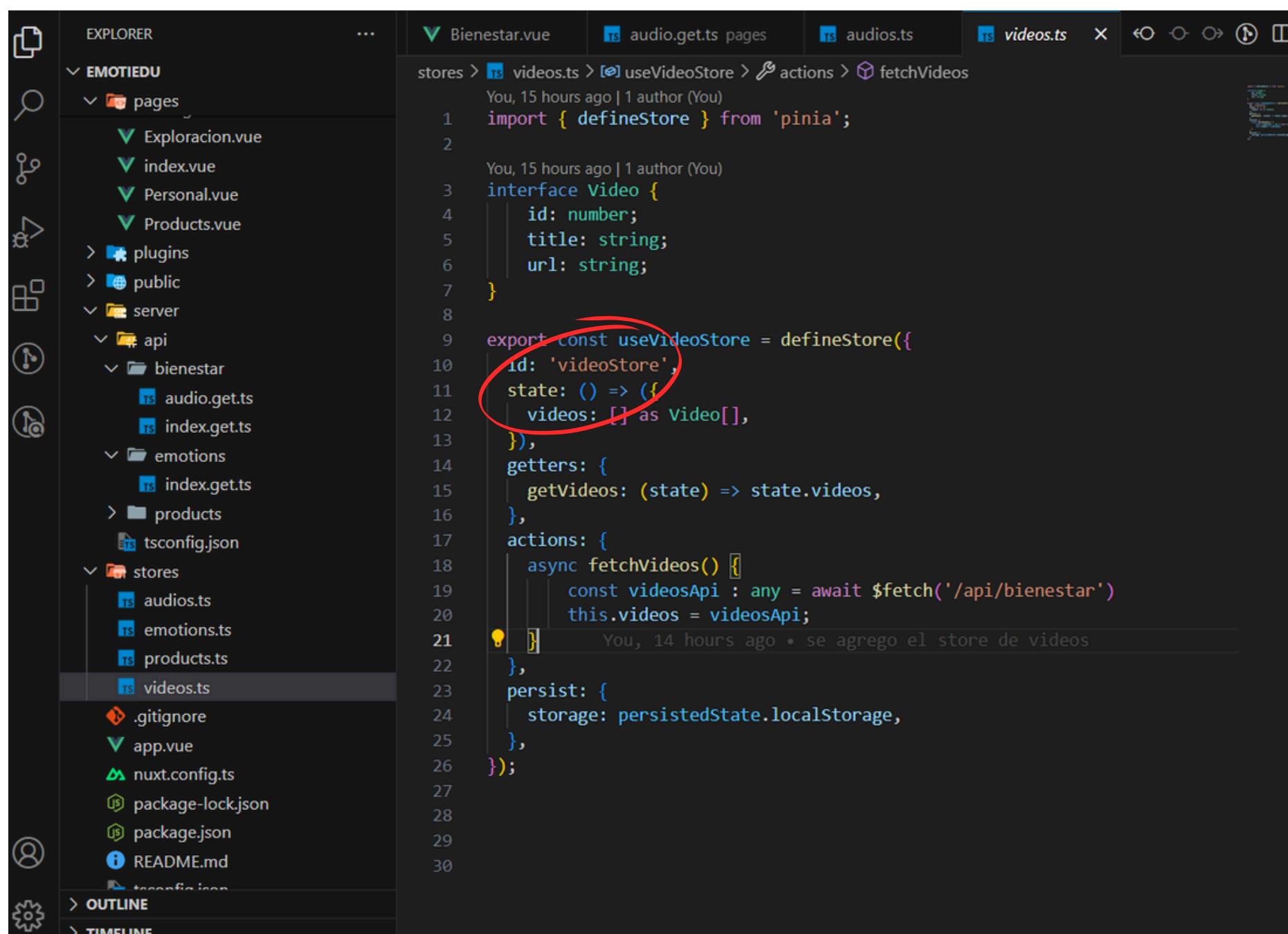


```
[{"title": "Conflictos y Heridas Emocionales", "audioUrl": "\u003ciframe style=\\"border-radius:12px\\" src=\\"https://open.spotify.com/embed/episode/5LaICDehMILTiC1s0Tioh7?utm_source=generator\\" width=\\"100%\\\" height=\\"152\\\" frameBorder=\\"0\\\" allowfullscreen=\\"\\\" allow=\\"autoplay; clipboard-write; encrypted-media; fullscreen; picture-in-picture\\\" loading=\\"lazy\\\"\u003e\u003c/iframe\u003e"}, {"title": "Poner L\u00edmites", "audioUrl": "\u003ciframe style=\\"border-radius:12px\\" src=\\"https://open.spotify.com/embed/episode/4y1laSf0GvCRPVFI2W09q?utm_source=generator\\" width=\\"100%\\\" height=\\"152\\\" frameBorder=\\"0\\\" allowfullscreen=\\"\\\" allow=\\"autoplay; clipboard-write; encrypted-media; fullscreen; picture-in-picture\\\" loading=\\"lazy\\\"\u003e\u003c/iframe\u003e"}, {"title": "Acompa\u00f1amiento Emocional", "audioUrl": "\u003ciframe style=\\"border-radius:12px\\" src=\\"https://open.spotify.com/embed/episode/1JQdiQpxxkNw7XQLWGeUkP?utm_source=generator\\" width=\\"100%\\\" height=\\"152\\\" frameBorder=\\"0\\\" allowfullscreen=\\"\\\" allow=\\"autoplay; clipboard-write; encrypted-media; fullscreen; picture-in-picture\\\" loading=\\"lazy\\\"\u003e\u003c/iframe\u003e"}]
```

Estos datos son enviados a la aplicación para ser procesados y utilizados.



3. Actualizar el State de la aplicación: Una vez que la aplicación recibe los datos de la API, el siguiente paso es actualizar el state con la nueva información recibida.

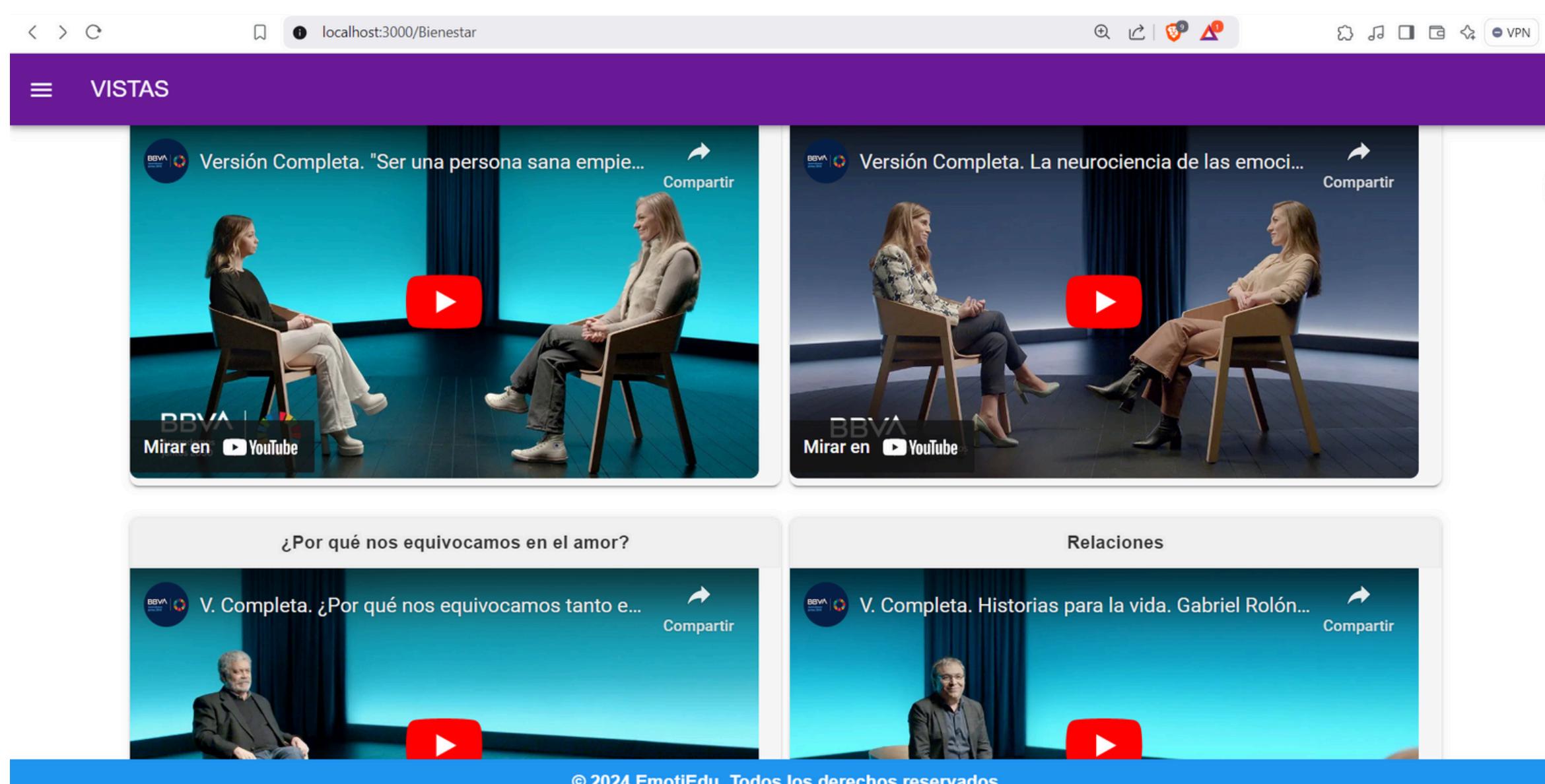


```
Bienestar.vue    audio.get.ts    audios.ts    videos.ts
stores > videos.ts > useVideoStore > actions > fetchVideos
You, 15 hours ago | 1 author (You)
import { defineStore } from 'pinia';

interface Video {
  id: number;
  title: string;
  url: string;
}

export const useVideoStore = defineStore({
  id: 'videoStore',
  state: () => ({
    videos: [] as Video[],
  }),
  getters: {
    getVideos: (state) => state.videos,
  },
  actions: {
    async fetchVideos() {
      const videosApi: any = await $fetch('/api/bienestar')
      this.videos = videosApi;
    },
    persist: {
      storage: persistedState.localStorage,
    },
  };
})
```

4. Actualizar la UI a partir del State: Al cambiar el state, la aplicación debe volver a renderizar su interfaz de usuario para reflejar los cambios. El state es el motor que determina qué se muestra en la pantalla.



Nota: Este proyecto está en constante evolución y sujeto a modificaciones, ya que se encuentra en una etapa de desarrollo continuo.

REFERENCIAS

<https://pinia.vuejs.org/>

<https://vuetifyjs.com/en/>

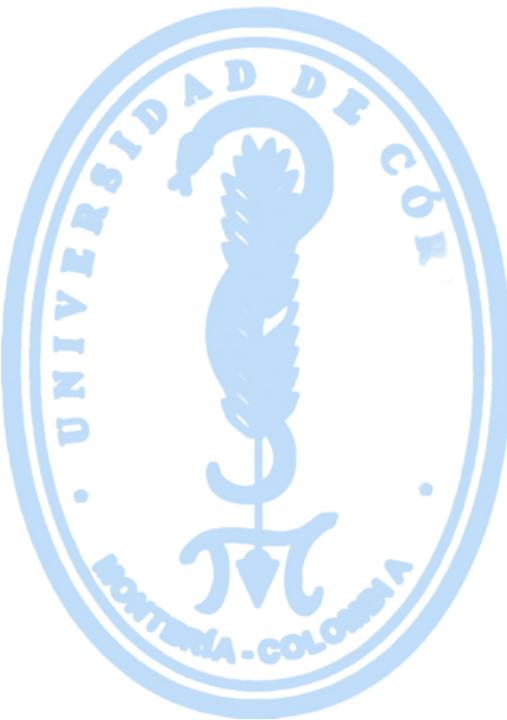
<https://vuejs.org/guide/components/props.html>

<https://jsoneditoronline.org/>

<https://www.postman.com/>

LICENCIATURA EN
INFORMÁTICA

Acreditada de Alta Calidad
MEN Res. 10710 25/05/17



EMOTIEDU