

Cronómetro educativo

28 de Noviembre del 2024

Integrantes del equipo

- Danilo José Ortega Sibaja
- Sebastián Andres Iozano Ramos

Explicación general

En el contexto educativo moderno, el uso de herramientas digitales se ha vuelto esencial para mejorar la experiencia de aprendizaje. Una de estas herramientas es el **cronómetro educativo**, una aplicación diseñada para optimizar el tiempo de estudio, seguimiento de actividades académicas, y la gestión de tareas asignadas a los estudiantes.

El cronómetro educativo desarrollado en este proyecto tiene como objetivo proporcionar una plataforma interactiva que permite a los usuarios (estudiantes) medir el tiempo dedicado a diferentes actividades académicas, registrar sus progresos y visualizar un historial detallado de sus sesiones de estudio. Al mismo tiempo, ofrece un **panel de administración** donde el profesor o administrador puede monitorear el desempeño de todos los estudiantes, asignar nuevas actividades y gestionar recursos educativos.

Datos clave

- El manejo efectivo del tiempo es fundamental para el éxito académico, ya que permite a los estudiantes organizar sus actividades de manera eficiente y equilibrar sus responsabilidades académicas y personales.
- La falta de organización del tiempo es una de las principales causas de estrés y bajo rendimiento en estudiantes universitarios, lo que resalta la necesidad de herramientas que promuevan una mejor planificación.
- La plataforma del **Cronómetro Educativo** ofrece una interfaz intuitiva que registra automáticamente el tiempo invertido en actividades académicas, fomentando hábitos de estudio consistentes y disciplina en la gestión del tiempo.

-
- El seguimiento de las actividades es personalizado para cada usuario, lo que permite a los estudiantes llevar un control independiente de su progreso, mientras que el administrador puede monitorear el desempeño colectivo desde un panel centralizado.
 - La inclusión de un sistema de asignación de actividades personalizadas por parte del administrador permite ofrecer recursos educativos adicionales, exámenes, y tareas específicas, adaptadas a las necesidades de cada estudiante.
 - El acceso a registros históricos de estudio brinda a los estudiantes una visión clara de sus patrones de estudio, permitiéndoles identificar áreas de mejora y establecer metas más realistas para optimizar su aprendizaje.

Objetivos

Objetivo General:

Desarrollar una plataforma digital que permita a los estudiantes gestionar de manera efectiva su tiempo de estudio mediante un cronómetro educativo, fomentando la organización, la disciplina en la gestión del tiempo y el seguimiento del progreso académico, con acceso a asignaciones personalizadas y recursos educativos adicionales.

Objetivos Específicos:

1. **Implementar un sistema de registro e inicio de sesión** que permita a los estudiantes acceder de manera segura y almacenar sus datos de progreso de forma persistente.
2. **Desarrollar una interfaz de cronómetro intuitiva** que registre automáticamente el tiempo invertido en actividades académicas y permita al usuario visualizar su historial de estudio.
3. **Crear un panel de administración para supervisores** donde puedan visualizar los registros de tiempo de todos los usuarios, asignar nuevas actividades, y gestionar el progreso académico de manera global.
4. **Ofrecer una funcionalidad de asignación de actividades personalizadas**, que permita al administrador asignar exámenes, juegos educativos, o recursos específicos a cada estudiante según sus necesidades.
5. **Garantizar la persistencia de datos** de usuarios y registros, permitiendo que cada estudiante mantenga su historial independiente, incluso después de reiniciar la aplicación o el servidor.
6. **Proporcionar un mensaje claro de “SIN ACTIVIDADES PENDIENTES”** cuando no existan tareas asignadas, para mantener una comunicación transparente con el usuario sobre el estado de sus responsabilidades académicas.

-
7. **Fomentar la autogestión del tiempo y el hábito de registrar actividades** como una herramienta para mejorar el rendimiento académico y la autoconciencia sobre el uso del tiempo.

Descripción del proyecto

El **Cronómetro Educativo** es una plataforma digital diseñada para ayudar a los estudiantes a gestionar de manera efectiva su tiempo de estudio, registrar sus actividades académicas y mejorar su rendimiento mediante un seguimiento continuo. La aplicación está organizada en diferentes componentes que ofrecen una experiencia de usuario intuitiva y funcional. A continuación, se describen los elementos clave de la plataforma:

Registro e Inicio de Sesión

Los usuarios deben registrarse para acceder a la plataforma mediante un formulario que solicita información básica, como nombre, apellido, edad, sexo, nombre de usuario e identificación. Una vez registrado, el usuario puede iniciar sesión utilizando sus credenciales.

- **Persistencia de Datos:** Los usuarios registrados y sus datos son almacenados en el servidor, lo que garantiza que la información se mantenga disponible incluso después de cerrar o reiniciar la aplicación.

Página Principal del Usuario

Tras iniciar sesión, los usuarios son dirigidos a una página personalizada que contiene:

- **Cronómetro Automático:** Un cronómetro que inicia automáticamente al acceder, registrando el tiempo dedicado a las actividades académicas hasta que el usuario cierra sesión.
- **Registro de Actividades:** Un componente que permite visualizar y gestionar el historial de tiempos registrados durante cada sesión de estudio.
- **Gestión de Actividades:** Si hay actividades asignadas por el administrador, estas se muestran en un listado. Si no hay actividades, el sistema muestra el mensaje "**SIN ACTIVIDADES PENDIENTES**".

Panel de Administración

El administrador, con credenciales exclusivas (User: **DniloADM** / Password: **11337541**), accede a un panel de control dedicado, desde donde puede:

-
- **Monitorear el Progreso:** Visualizar los registros de tiempo de todos los usuarios, de forma individual o global, permitiendo identificar el desempeño de cada estudiante.
 - **Asignar Actividades:** Agregar o gestionar recursos educativos, como juegos, exámenes o tareas específicas, para uno o varios estudiantes.
 - **Supervisar la Actividad Global:** Evaluar la participación de los estudiantes en las actividades asignadas y tomar decisiones informadas para mejorar el proceso de enseñanza.

Estructura General de la Plataforma

- **Interfaz de Usuario Intuitiva:** La aplicación está diseñada para ser fácil de usar tanto para estudiantes como para el administrador, con una navegación clara y componentes visuales simples pero efectivos.
- **Sincronización en Tiempo Real:** La aplicación actualiza y sincroniza automáticamente los datos de tiempo y actividades con el servidor, proporcionando una experiencia fluida para el usuario.
- **Accesibilidad y Personalización:** Cada usuario tiene una experiencia personalizada basada en su cuenta, lo que permite un control individual sobre su progreso y actividades.

El **Cronómetro Educativo** no solo es una herramienta para la gestión del tiempo, sino que también es un recurso valioso para fomentar la autogestión y la disciplina en los estudiantes, mientras que facilita al administrador el seguimiento y la mejora continua del proceso educativo.

Hitos

A continuación, se presenta un esquema detallado de la estructura y los componentes clave del **Cronómetro Educativo**. Este boceto define la organización de las vistas principales, las funcionalidades interactivas, y la experiencia del usuario dentro de la aplicación:

VISTA PRINCIPAL

Componente de Inicio de Sesión y Registro

- **Formulario de Inicio de Sesión:**
 - Campos:
 - Usuario (name-user)
 - Contraseña

-
- Botón: **Iniciar Sesión**
 - Enlace: **Registrarse**, que redirige al formulario de registro.

Formulario de Registro

Este formulario aparece al seleccionar la opción de "Registrarse" y solicita:

- Nombres
- Apellidos
- Edad
- Sexo
- Identificación
- Nombre de Usuario (name-user)
- Contraseña
- Botón: **Registrar**

Una vez registrado, el usuario será redirigido a la vista principal con su cronómetro activado.

PÁGINA PRINCIPAL DEL USUARIO

Barra de Navegación

- **Menú de Navegación** con las siguientes opciones:
 - **Inicio**
 - **Historial de Actividades**
 - **Cerrar Sesión**
- **Avatar del Usuario** que despliega la configuración de perfil.
- **Indicador del Cronómetro Activo** en la esquina superior derecha.

Componente de Cronómetro

- **Visualización del Tiempo:**
 - Pantalla que muestra el tiempo transcurrido en formato **hh:mm:ss**.

Componente de Registro de Actividades

- Lista cronológica de los registros guardados por el usuario.

-
- Fecha y hora de inicio y fin de cada sesión.
 - Duración total de la sesión.
 - Botón para eliminar registros específicos si es necesario.

VISTA DEL ADMINISTRADOR

Panel de Control del Administrador

Accesible únicamente con el usuario: **xxxxxx** y la contraseña **xxxxxx**.

Componente de Visualización de Registros

- Lista completa de registros de todos los usuarios.
 - Nombre de Usuario
 - Identificación
 - Tiempo Total Acumulado
 - Número de Actividades Completadas

Componente de Gestión de Actividades

- Opción para **asignar nuevas actividades** a usuarios específicos.
 - Campo de texto para descripción de la actividad.
 - Botón: **Asignar Actividad**.

Gestión de Recursos

- Enlace a diferentes recursos educativos (juegos, exámenes, etc.) visibles desde la barra de navegación.
- Opción para editar o eliminar recursos asignados.

Mensajes de Estado

- Si no hay actividades asignadas, se muestra el mensaje: **"SIN ACTIVIDADES PENDIENTES"** en la pantalla principal del usuario.

Componente de Configuración de Perfil

- Cambio de **avatar** del usuario.
- Modificación de **información personal** (nombre, apellidos, contraseña).

- **Modo oscuro:** opción para activar o desactivar el modo oscuro en toda la aplicación.

Componente Cerrar Sesión

- Botón en la barra de navegación que permite al usuario finalizar su sesión actual y dirigirse a la página de inicio de sesión.

Flujo General

1. **Inicio de Sesión o Registro del Usuario.**
2. **Acceso a la Página Principal** con el cronómetro activado automáticamente.
3. **Registro Automático del Tiempo** hasta que el usuario cierre sesión.
4. El **Administrador** puede monitorear y gestionar el progreso desde el panel de control.
5. El usuario puede acceder a su historial, ver asignaciones o consultar recursos educativos asignados.

Estructura del código

Componentes:

```
1 <template>
2 <div v-if="!isLoggedIn" class="overlay">
3 <div class="form-container">
4 <h2>{{ isLoggedIn ? 'Iniciar Sesión' : 'Registrarse' }}</h2>
5
6 <!-- Formulario de inicio de sesión -->
7 <form v-if="isLoggedIn" @submit.prevent="login">
8 <input v-model="username" type="text" placeholder="Usuario" required />
9 <input v-model="password" type="password" placeholder="Contraseña" required />
10 <button type="submit">Ingresar</button>
11 </form>
12
13 <!-- Formulario de registro -->
14 <form v-if="!isLoggedIn" @submit.prevent="register">
15 <input v-model="newUsername" type="text" placeholder="Nombre de usuario" required />
16 <input v-model="newPassword" type="password" placeholder="Contraseña" required />
17 <input v-model="firstName" type="text" placeholder="Nombres" required />
18 <input v-model="lastName" type="text" placeholder="Apellidos" required />
19 <input v-model="age" type="number" placeholder="Edad" required />
20 <select v-model="gender" required>
21 <option value="" disabled>Sexo</option>
22 <option value="M">Masculino</option>
23 <option value="F">Femenino</option>
24 </select>
25 <input v-model="identification" type="text" placeholder="Identificación" required />
26 <button type="submit">Registrar</button>
27 </form>
28
29 <!-- Enlace para cambiar entre login y registro -->
30 <p>
```

Archivo Editar Selección Ver Ir ... paul_cronos [Administrador]

EXPLORADOR

- PAUL_CRONOS
 - .idx
 - .vscode
 - components
 - LoginForm.vue
 - RegisterForm.vue
 - TimerButtons.vue
 - TimerDisplay.vue.vue
 - UserInfo.vue
 - layout
 - pages
 - public
 - plugins
 - server
 - .gitignore
 - app.vue
 - nuxt.config.ts
 - package-lock.json
 - package.json
 - README.md
 - tailwind.config.js
 - tsconfig.json
- ESQUEMA
- LÍNEA DE TIEMPO

RegisterForm.vue

```
1 <template>
2   <div class="overlay">
3     <div class="form-container">
4       <h2>Registrar Usuario</h2>
5       <input v-model="firstName" type="text" placeholder="Nombres" required />
6       <input v-model="lastName" type="text" placeholder="Apellidos" required />
7       <input v-model="age" type="number" placeholder="Edad" required />
8       <select v-model="gender" required>
9         <option value="">Seleccione Sexo</option>
10        <option value="Masculino">Masculino</option>
11        <option value="Femenino">Femenino</option>
12      </select>
13      <input v-model="idNumber" type="text" placeholder="Identificación" required />
14      <input v-model="username" type="text" placeholder="Nombre de Usuario" required />
15      <input v-model="password" type="password" placeholder="Contraseña" required />
16      <button @click="register">Registrar</button>
17      <button @click="$emit('close')">Cancelar</button>
18    </div>
19  </div>
20 </template>
21
22 <script setup>
23   import { ref } from 'vue';
24
25   const firstName = ref('');
26   const lastName = ref('');
27   const age = ref('');
28   const gender = ref('');
29   const idNumber = ref('');
30   const username = ref('');
```

main 0 0 0 BLACKBOX Chat Add Logs Improve Code Share Code Link Generate Commit Message Espacios: 2 UTF-8 CRLF Vue AI Code Chat Prettier

Buscar

Noche lluviosa 5:22 p. m. 28/11/2024

Archivo Editar Selección Ver Ir ... paul_cronos [Administrador]

EXPLORADOR

- PAUL_CRONOS
 - .idx
 - .vscode
 - components
 - LoginForm.vue
 - RegisterForm.vue
 - TimerButtons.vue
 - TimerDisplay.vue.vue
 - UserInfo.vue
 - layout
 - pages
 - public
 - plugins
 - server
 - .gitignore
 - app.vue
 - nuxt.config.ts
 - package-lock.json
 - package.json
 - README.md
 - tailwind.config.js
 - tsconfig.json
- ESQUEMA
- LÍNEA DE TIEMPO

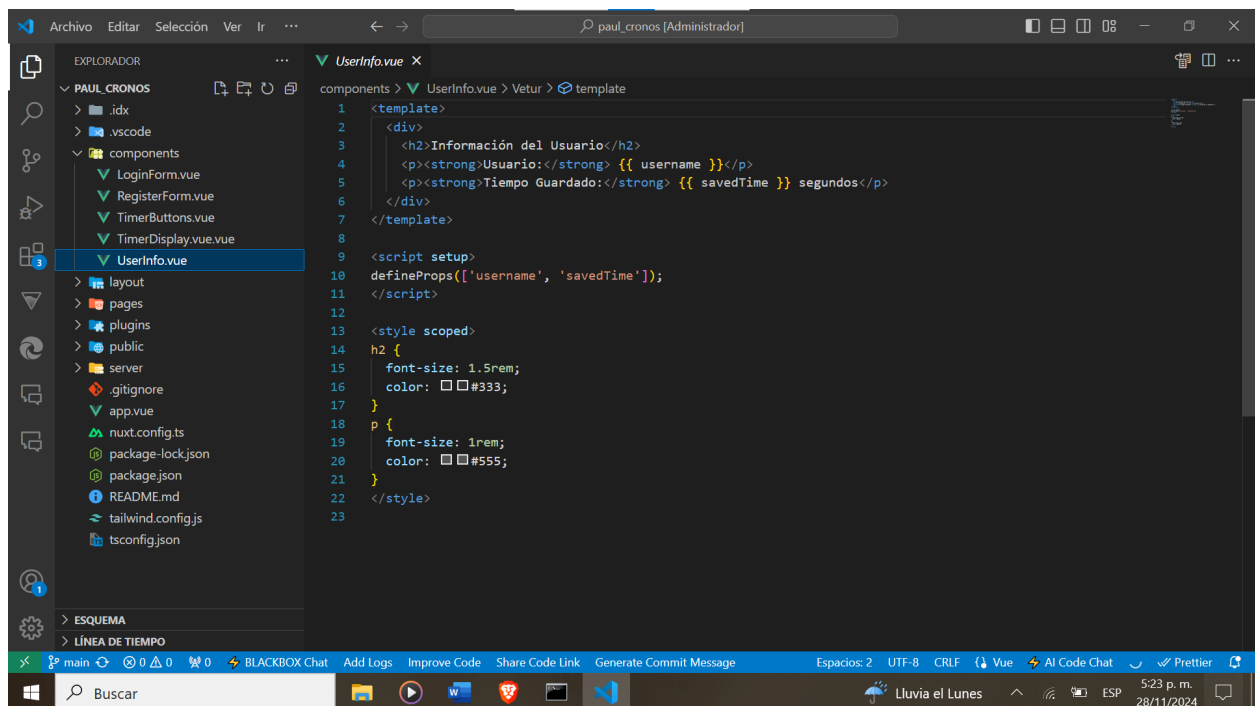
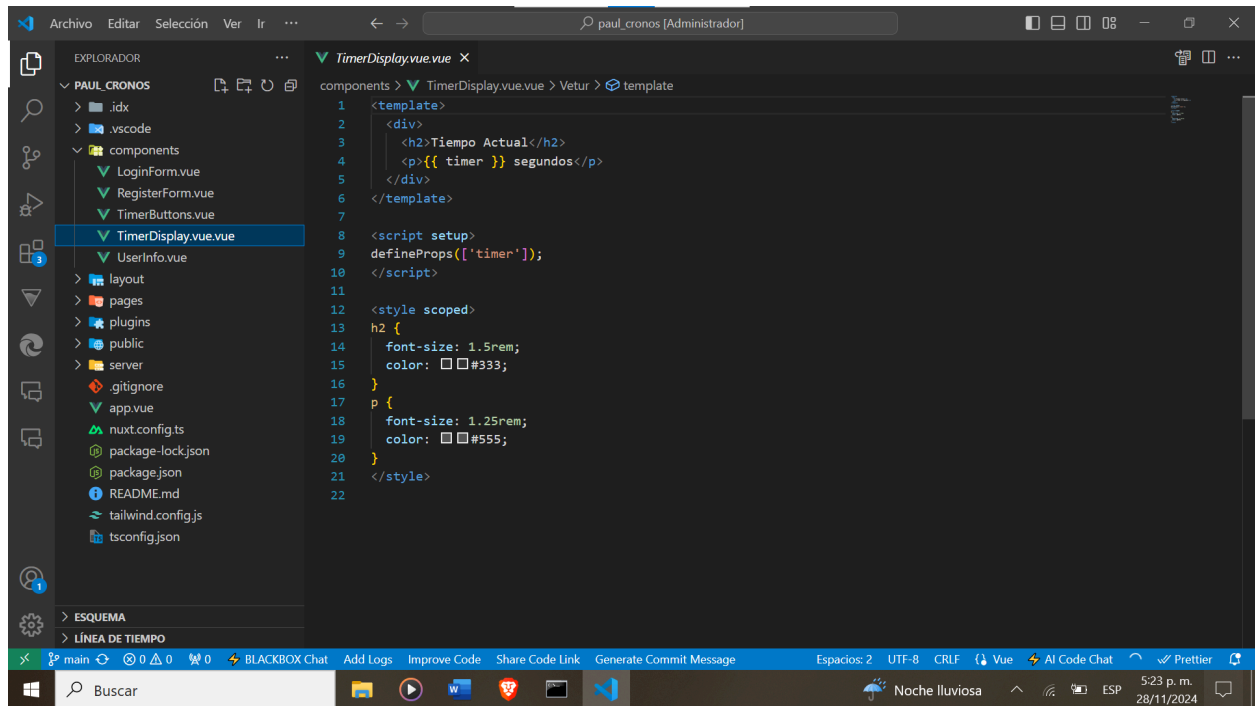
TimerButtons.vue

```
1 <template>
2   <div>
3     <button @click="startTimer">Iniciar</button>
4     <button @click="saveTime">Guardar</button>
5   </div>
6 </template>
7
8 <script setup>
9   defineEmits(['start', 'save']);
10
11   const startTimer = () => {
12     emit('start');
13   };
14
15   const saveTime = () => {
16     emit('save');
17   };
18 </script>
19
20 <style scoped>
21   button {
22     margin: 0 10px;
23     padding: 10px 20px;
24     font-size: 1rem;
25     border: none;
26     border-radius: 5px;
27     cursor: pointer;
28   }
29   button:first-of-type {
30     background-color: #4caf50;
```

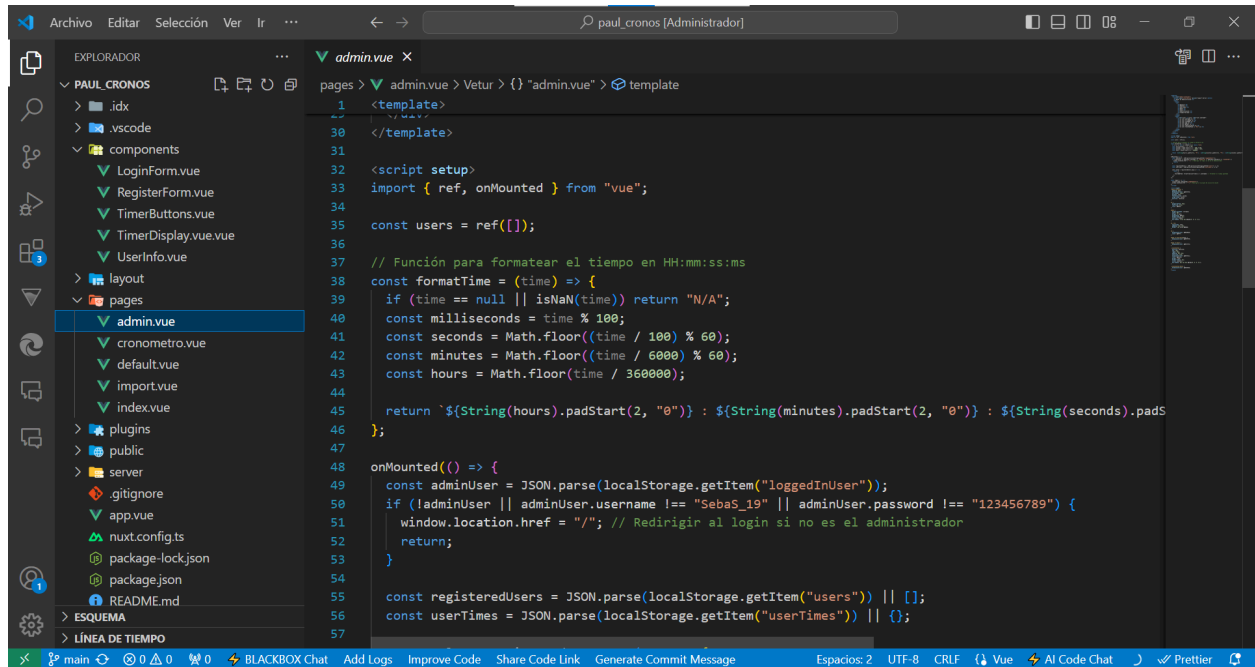
main 0 0 0 BLACKBOX Chat Add Logs Improve Code Share Code Link Generate Commit Message Espacios: 2 UTF-8 CRLF Vue AI Code Chat Prettier

Buscar

Noche lluviosa 5:22 p. m. 28/11/2024

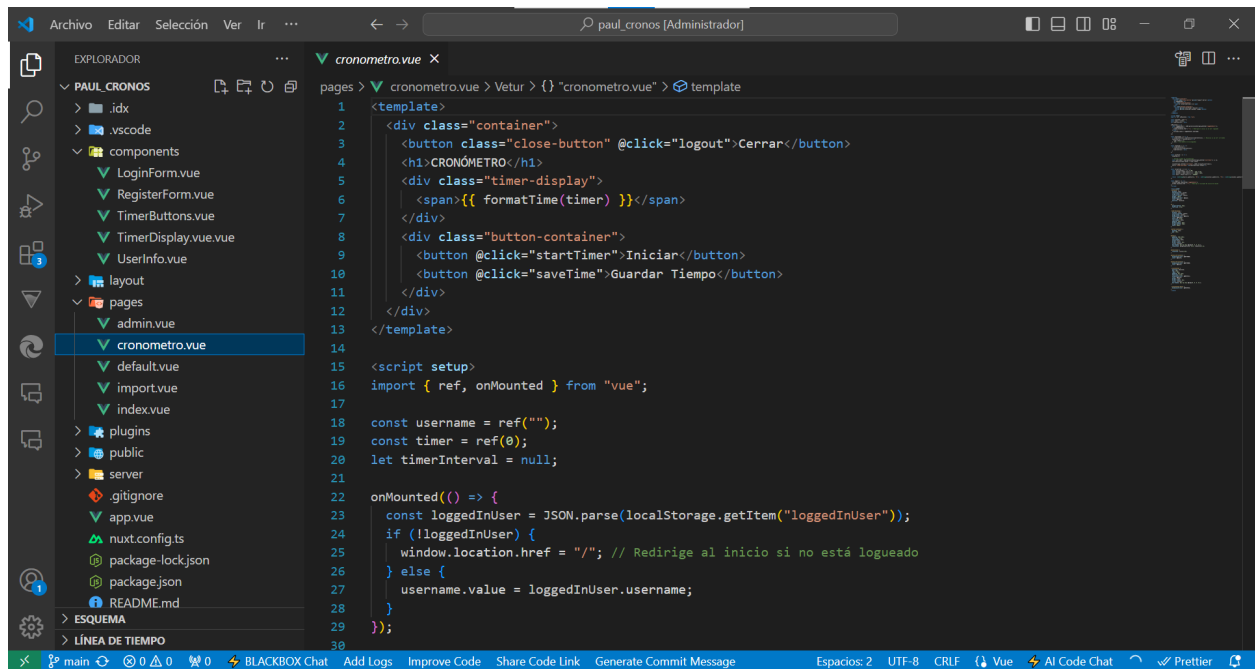


Pages



The screenshot shows the VS Code editor with the file explorer on the left. The project structure includes a 'pages' directory under 'PAUL_CRONOS'. The 'admin.vue' file is selected in the explorer and its content is displayed in the editor. The code defines a Vue component for the admin page, including a template, script setup, and logic for user authentication and time formatting.

```
1 <template>
2 </template>
3
4 <script setup>
5 import { ref, onMounted } from "vue";
6
7 const users = ref([]);
8
9 // Función para formatear el tiempo en HH:mm:ss:ms
10 const formatTime = (time) => {
11   if (time == null || isNaN(time)) return "N/A";
12   const milliseconds = time % 100;
13   const seconds = Math.floor((time / 100) % 60);
14   const minutes = Math.floor((time / 6000) % 60);
15   const hours = Math.floor(time / 360000);
16
17   return `${String(hours).padStart(2, "0")}:${String(minutes).padStart(2, "0")}:${String(seconds).padStart(2, "0")}:${String(milliseconds).padStart(2, "0")}`;
18 };
19
20 onMounted(() => {
21   const adminUser = JSON.parse(localStorage.getItem("loggedInUser"));
22   if (!adminUser || adminUser.username !== "SebaS_19" || adminUser.password !== "123456789") {
23     window.location.href = "/"; // Redirigir al login si no es el administrador
24     return;
25   }
26
27   const registeredUsers = JSON.parse(localStorage.getItem("users")) || [];
28   const userTimes = JSON.parse(localStorage.getItem("userTimes")) || [];
29 });
```



The screenshot shows the VS Code editor with the file explorer on the left. The project structure includes a 'pages' directory under 'PAUL_CRONOS'. The 'cronometro.vue' file is selected in the explorer and its content is displayed in the editor. The code defines a Vue component for the timer page, including a template, script setup, and logic for user authentication and timer functionality.

```
1 <template>
2 <div class="container">
3   <button class="close-button" @click="logout">Cerrar</button>
4   <h1>CRONÓMETRO</h1>
5   <div class="timer-display">
6     <span>{{ formatTime(timer) }}</span>
7   </div>
8   <div class="button-container">
9     <button @click="startTimer">Iniciar</button>
10    <button @click="saveTime">Guardar Tiempo</button>
11  </div>
12 </div>
13 </template>
14
15 <script setup>
16 import { ref, onMounted } from "vue";
17
18 const username = ref("");
19 const timer = ref(0);
20 let timerInterval = null;
21
22 onMounted(() => {
23   const loggedInUser = JSON.parse(localStorage.getItem("loggedInUser"));
24   if (!loggedInUser) {
25     window.location.href = "/"; // Redirige al inicio si no está logueado
26   } else {
27     username.value = loggedInUser.username;
28   }
29 });
30
```

Archivo Editar Selección Ver Ir ... paul_cronos [Administrador]

EXPLORADOR

- PAUL_CRONOS
 - .idx
 - .vscode
 - components
 - LoginForm.vue
 - RegisterForm.vue
 - TimerButtons.vue
 - TimerDisplay.vue.vue
 - UserInfo.vue
 - layout
 - pages
 - admin.vue
 - cronometro.vue
 - default.vue
 - import.vue
 - index.vue
 - plugins
 - public
 - server
 - .gitignore
 - app.vue
 - nuxt.config.ts
 - package-lock.json
 - package.json
 - README.md
- ESQUEMA
- LÍNEA DE TIEMPO

import.vue X

```
pages > import.vue > Vetur > {} "import.vue" > template
1 <template>
2   <div>
3     <h1 class="text-3xl font-bold text-center my-4">Bienvenido al Sistema de Cronómetro</h1>
4     <!-- Aquí se inserta el Panel de Administración -->
5     <AdminPanel />
6   </div>
7 </template>
8
9 <script>
10 // Importar el componente AdminPanel
11 import AdminPanel from '~/components/AdminPanel.vue';
12
13 export default {
14   components: {
15     AdminPanel,
16   },
17 };
18 </script>
19
```

main 0 0 0 BLACKBOX Chat Add Logs Improve Code Share Code Link Generate Commit Message Espacios: 2 UTF-8 CRLF Vue AI Code Chat Prettier

Archivo Editar Selección Ver Ir ... paul_cronos [Administrador]

EXPLORADOR

- PAUL_CRONOS
 - .idx
 - .vscode
 - components
 - LoginForm.vue
 - RegisterForm.vue
 - TimerButtons.vue
 - TimerDisplay.vue.vue
 - UserInfo.vue
 - layout
 - pages
 - admin.vue
 - cronometro.vue
 - default.vue
 - import.vue
 - index.vue
 - plugins
 - public
 - server
 - .gitignore
 - app.vue
 - nuxt.config.ts
 - package-lock.json
 - package.json
 - README.md
- ESQUEMA
- LÍNEA DE TIEMPO

index.vue X

```
pages > index.vue > Vetur > {} "index.vue" > template
1 <template>
2   <div v-if="!isLoggedIn" class="overlay">
3     <div class="form-container">
7       <form v-if="isLogin" @submit.prevent="login">
11       </form>
12
13       <!-- Formulario de registro -->
14       <form v-if="!isLogin" @submit.prevent="register">
15         <input v-model="newUsername" type="text" placeholder="Nombre de usuario" required />
16         <input v-model="newPassword" type="password" placeholder="Contraseña" required />
17         <input v-model="firstName" type="text" placeholder="Nombres" required />
18         <input v-model="lastName" type="text" placeholder="Apellidos" required />
19         <input v-model="age" type="number" placeholder="Edad" required />
20         <select v-model="gender" required>
21           <option value="" disabled>Sexo</option>
22           <option value="M">Masculino</option>
23           <option value="F">Femenino</option>
24         </select>
25         <input v-model="identification" type="text" placeholder="Identificación" required />
26         <button type="submit">Registrar</button>
27       </form>
28
29       <!-- Enlace para cambiar entre Iniciar sesión y Registrarse -->
30       <p>
31         <span v-if="isLogin">
32           ¿No tienes cuenta? <a href="#" @click="toggleForm">Crea una cuenta</a>
33         </span>
34         <span v-if="!isLogin">
35           ¿Ya tienes cuenta? <a href="#" @click="toggleForm">Inicia sesión</a>
36         </span>
37       </p>
38     </div>
39   </div>
40 </template>
```

main 0 0 0 BLACKBOX Chat Add Logs Improve Code Share Code Link Generate Commit Message Espacios: 2 UTF-8 CRLF Vue AI Code Chat Prettier

Vista previa

Inicio



The image shows a preview of a login form. The form is centered on a dark gray background. It has a white background with rounded corners. At the top, the title 'Iniciar Sesión' is displayed in a bold, black, serif font. Below the title are two input fields: the first is labeled 'Usuario' and the second is labeled 'Contraseña'. Both labels are in a gray, sans-serif font. Below the input fields is a button labeled 'Ingresar' in a gray, sans-serif font. Below the button is a link that says '¿No tienes cuenta? [Crea una cuenta](#)' in a gray, sans-serif font. At the bottom of the form is a red button labeled 'Cerrar' in a white, sans-serif font.

Iniciar Sesión

¿No tienes cuenta? [Crea una cuenta](#)

Registrarse



¿Ya tienes cuenta? [Inicia sesión](#)

Apartado del estudiante

Cerrar

CRONÓMETRO

00 : 00 : 00 : 00

Iniciar Guardar Tiempo

Apartado del administrador

Cerrar

Panel de Administración

Usuario	Nombres	Apellidos	Edad	Sexo	Identificación	Tiempo Marcado
sebas	s	l	45	M	3353646	00 : 00 : 05 : 37
sebas1	3	3	45	M	45656	00 : 00 : 02 : 18
sebas2	d	s	34	M	34567	00 : 01 : 04 : 83

Enlace del repositorio:

https://github.com/Grupo-Investigacion-Bimadino/paul_cronos