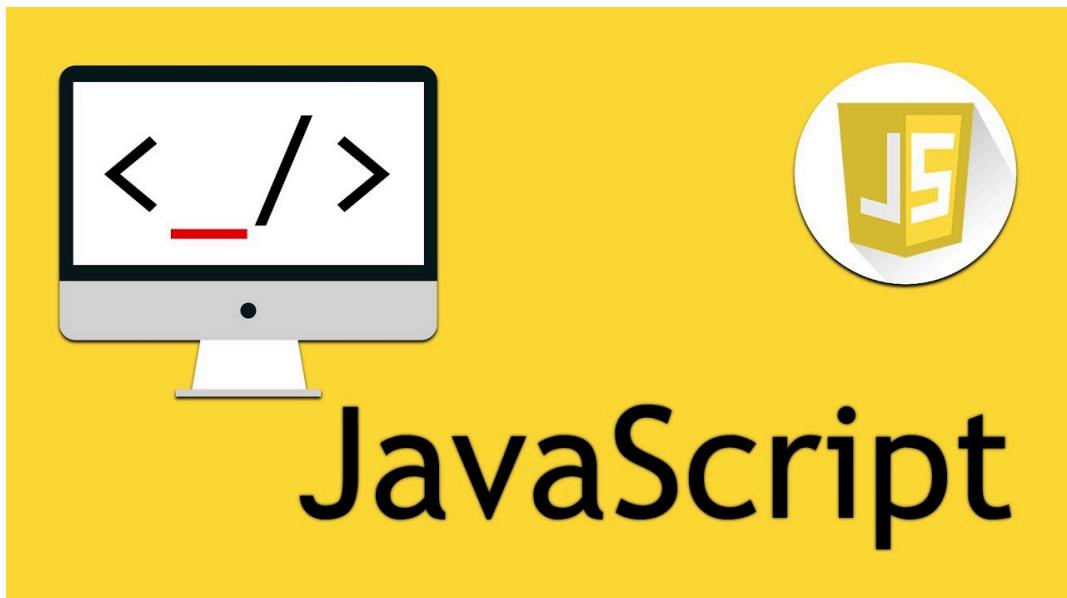


Projecte JavaScript



Professor: Jordi Vega
Assignatura: M7 Desenvolupament d'interfícies
Nom: Manuel Steven
Cognoms: Romero Vargas

Índex

Documentació del procediment	3
Fitxers generals	3
Index.html	3
Vite.config.js	3
Package.json	4
Style.css	4
Main.js	5
App.vue	6
Index.js	7
Components	8
AppHeader.vue	8
AppFooter.vue	8
Vistes	9
LandingPage.vue	9
CalculatorPage.vue	10
CommentsPage.vue	13
UsersPage.vue	14
Tests	15
Calculator.test.js	15
Contingut	15
Prova	15
Comments.test.js	16
Contingut	16
Prova	16
Users.test.js	17
Contingut	17
Prova	17
Resultat	18
Header Responsiu	18
LandingPage	18
Pantalla completa	18
Responsive	19
CalculatorPage	20
Pantalla completa	20
Responsive	20
CommentsPage	21
Pantalla completa	21
Responsive	21
UsersPage	22
Pantalla Completa	22

Documentació del procediment

Fitxers generals

Index.html

En el index.html només he canviat la ruta al logo de la pàgina per a que tingui un propi.

```
↳ index.html ×
↳ index.html > ⏺ html
1   <!doctype html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8" />
5       <link rel="icon" type="image/svg+xml" href="/logo.jpg" />
6       <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7       <title>Projecte JavaScript</title>
8       <link rel="stylesheet" href="/src/style.css" />
9     </head>
10    <body>
11      <div id="app"></div>
12      <script type="module" src="/src/main.js"></script>
13    </body>
14  </html>
```

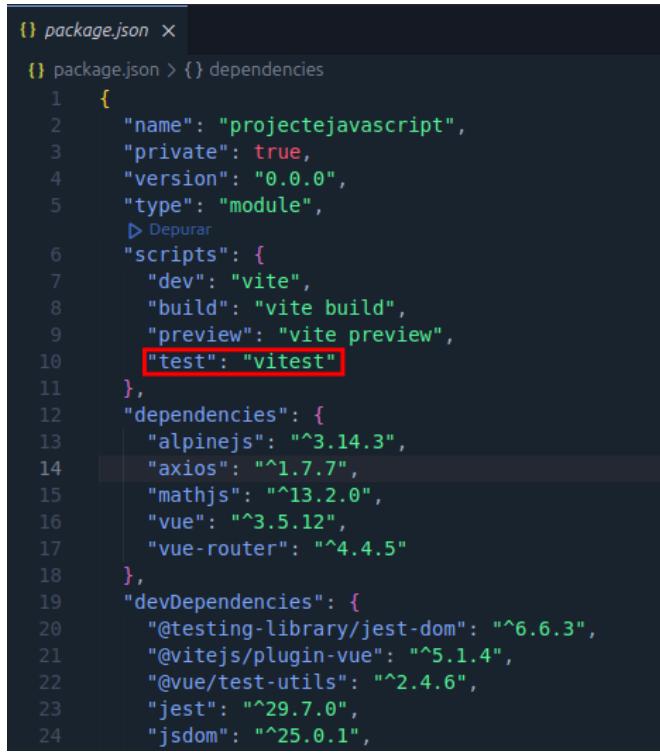
Vite.config.js

En l'arxiu de configuració de vite haurem de confirmar que tinguem el alias @ per a la carpeta **src**, ademés haurem d'excluir dels testos la carpeta **node_modules** i **dist**

```
js vite.config.js ×
js vite.config.js > [e] default
1 // vite.config.js
2 import { defineConfig } from 'vite';
3 import vue from '@vitejs/plugin-vue';
4 import path from 'path';
5
6 export default defineConfig({
7   plugins: [vue()],
8   resolve: {
9     alias: {
10       '@': path.resolve(__dirname, 'src'), // alias '@' para la carpeta 'src'
11     },
12   },
13   test: {
14     globals: true,
15     environment: 'jsdom',
16     setupFiles: '@testing-library/jest-dom',
17     exclude: ['node_modules', 'dist'], // Excluir directorios comunes de las pruebas
18   },
19 });
```

Package.json

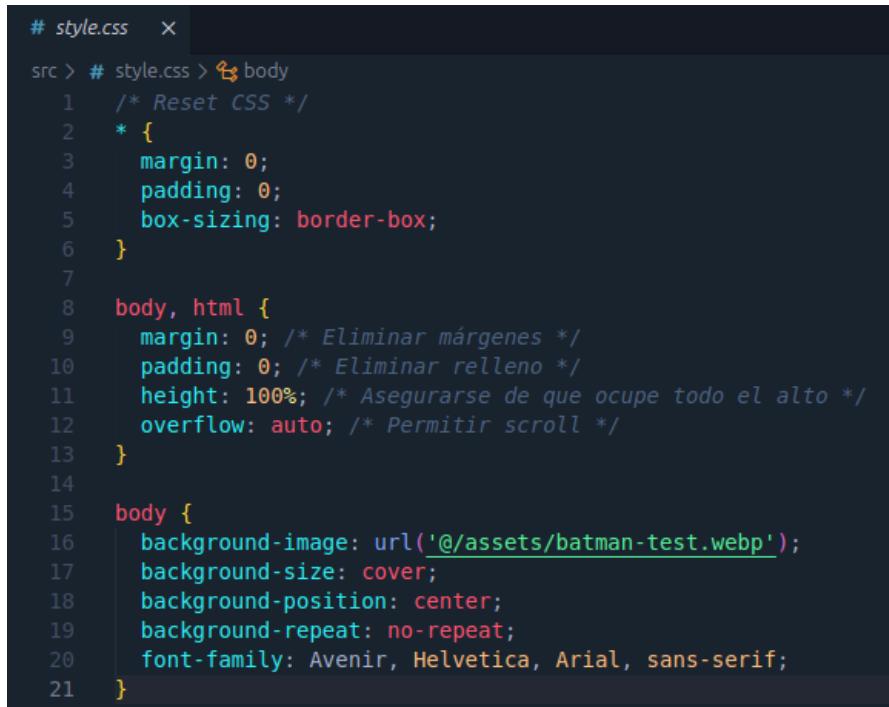
En el package.json, podrem comprovar les versions i dependencies dels paquets instal·lats en el nostre projecte, ademés he afegit en scripts, test que funcione com el **vitest**, que ens servirà per a comprovar el funcionament dels testos creats.



```
# package.json ×
{} package.json > {} dependencies
1 {
2   "name": "proyectejavascript",
3   "private": true,
4   "version": "0.0.0",
5   "type": "module",
6   ▶ Depurar
7   "scripts": {
8     "dev": "vite",
9     "build": "vite build",
10    "preview": "vite preview",
11    "test": "vitest"
12  },
13  "dependencies": {
14    "alpinejs": "^3.14.3",
15    "axios": "^1.7.7",
16    "mathjs": "^13.2.0",
17    "vue": "^3.5.12",
18    "vue-router": "^4.4.5"
19  },
20  "devDependencies": {
21    "@testing-library/jest-dom": "^6.6.3",
22    "@vitejs/plugin-vue": "^5.1.4",
23    "@vue/test-utils": "^2.4.6",
24    "jest": "^29.7.0",
25    "jsdom": "^25.0.1",
26  }
}
```

Style.css

En l'arxiu d'estils tindrem els estils més bàsics i globals per a la pàgina, el més rellevant ha sigut afegir la imatge de fons per al projecte.



```
# style.css ×
src > # style.css > body
1  /* Reset CSS */
2  * {
3    margin: 0;
4    padding: 0;
5    box-sizing: border-box;
6  }
7
8  body, html {
9    margin: 0; /* Eliminar márgenes */
10   padding: 0; /* Eliminar relleno */
11   height: 100%; /* Asegurarse de que ocupe todo el alto */
12   overflow: auto; /* Permitir scroll */
13 }
14
15 body {
16   background-image: url('@/assets/batman-test.webp');
17   background-size: cover;
18   background-position: center;
19   background-repeat: no-repeat;
20   font-family: Avenir, Helvetica, Arial, sans-serif;
21 }
```

Main.js

En l'arxiu main.js els canvis fets han sigut importar js i afegir un codi javascript per a poder obrir i tancar un menú mòbil en el header.

```
JS main.js    X

src > JS main.js > ...
1  import { createApp } from 'vue';
2  import App from './App.vue';
3  import router from './router'; // Si estás usando Vue Router
4  import Alpine from 'alpinejs'
5  window.Alpine = Alpine
6  Alpine.start()
7  createApp(App)
8    .use(router)
9    .mount('#app');
10
11 document.addEventListener('DOMContentLoaded', () => {
12   const menuToggle = document.querySelector('.menu-toggle');
13   const navList = document.querySelector('.nav-list');
14
15   if (menuToggle) {
16     menuToggle.addEventListener('click', () => {
17       navList.classList.toggle('nav-list-mobile');
18       console.log('Menu toggled');
19     });
20   }
21 });
22
23 const app = createApp(App);
24 app.use(router);
25 app.mount('#app');
```

App.vue

En la pàgina de l'aplicació he importat els components del **Header** i el **Footer**.

```
▼ App.vue M X
src > ▼ App.vue > {} script
1  <template>
2    <div id="app">
3      <AppHeader />
4      <div class="content">
5        <router-view />
6      </div>
7      <AppFooter />
8    </div>
9  </template>
10 
11 <script>
12   import AppHeader from './components/AppHeader.vue';
13   import AppFooter from './components/AppFooter.vue';
14 
15   export default {
16     name: 'App',
17     components: {
18       AppHeader,
19       AppFooter,
20     },
21   };
22 </script>
```

A més a més he afegit uns padding que afectaran en general a totes les vistes de manera que puguin mantindre un marge en el header i en les vores laterals.

```
<style>
#app {
  display: flex;
  flex-direction: column;
  min-height: 100vh;
}

.content {
  flex: 1;
  padding-top: 6rem;
  padding-left: 3rem;
  padding-right: 3rem;
}
</style>
```

Index.js

En el directori **routes** haurem d'editar l'arxiu index per a importar i afegir les rutes per a cada vista.

```
js index.js M X
src > router > js index.js > [e] routes
1  // src/router/index.js
2  import { createRouter, createWebHistory } from 'vue-router';
3  import LandingPage from '@/views/LandingPage.vue';
4  import CalculatorPage from '@/views/CalculatorPage.vue';
5  import CommentsPage from '@/views/CommentsPage.vue';
6  import UsersPage from '@/views/UsersPage.vue';
7
8
9  const routes = [
10  {
11    path: '/',
12    name: 'LandingPage',
13    component: LandingPage,
14  },
15  {
16    path: '/calculator',
17    name: 'Calculator',
18    component: CalculatorPage,
19  },
20  {
21    path: '/comments',
22    name: 'Comments',
23    component: CommentsPage,
24  },
25  {
26    path: '/users',
27    name: 'Users',
28    component: UsersPage,
29  },

```

```
const router = createRouter({
  history: createWebHistory('/'), // Ajusta aquí la base
  routes,
});

export default router;
```

Components

AppHeader.vue

En el header tindrem la nostra barra de navegació en un menu mobil que ens funcionara al obrir la pagina en un mòbil

```
▼ AppHeader.vue ×
src > components > ▼ AppHeader.vue > {} template > ⚡ header.header > ⚡ nav.nav > ⚡ a.logo-container > ⚡ h1.app-title
1  <template>
2    <header class="header" x-data="{ isMenuOpen: false }">
3      <!-- Logo y Título -->
4      <a href="#" class="logo-container">
5        
6        <h1 class="app-title">Aplicació JavaScript</h1>
7      </a>
8
9
10     <!-- Botón del menú para móvil -->
11     <button
12       class="menu-toggle"
13       @click="isMenuOpen = !isMenuOpen; toggleMenu()"
14       aria-label="Toggle menu"
15     >
16       ≡
17     </button>
18
19     <!-- Menú de navegación -->
20     <ul :class="['nav-list': true, 'nav-list-mobile': isMenuOpen ]">
21       <li><router-link to="/">Inici</router-link></li>
22       <li><router-link to="/calculator">Calculadora</router-link></li>
23       <li><router-link to="/comments">Comentaris</router-link></li>
24       <li><router-link to="/users">Usuaris</router-link></li>
25     </ul>
26   </nav>
27 </header>
28 </template>
```

AppFooter.vue

```
▼ AppFooter.vue ×
src > components > ▼ AppFooter.vue > {} script > [?] default
1  <template>
2    <footer>
3      <p>&copy; 2024 Aplicació produïda Steven Inc.</p>
4    </footer>
5  </template>
6
7  <script>
8  export default {
9    name: 'AppFooter',
10  };
11 </script>
12
```

Vistes

LandingPage.vue

En la landing page simplement afegirem una simple portada que tindrà unes petites descripcions humildes de les funcions de cada pàgina estableerta, en un botó per a accedir a cada una-

```
▼ LandingPage.vue ×
src > views > ▼ LandingPage.vue > {} template > div.home-page > div.content-container > div.features > div.feature-card
1  <template>
2    <div class="home-page">
3      <div class="content-container">
4        <h1 class="title">Benvinguts a l'Aplicació JavaScript</h1>
5        <p class="description">Aquí pots trobar diverses eines útils:</p>
6        <div class="features">
7          <div class="feature-card">
8            <h2>Calculadora</h2>
9            <p>Realitza operacions matemàtiques de manera senzilla i ràpida.</p>
10           <router-link to="/calculator" class="feature-link">Prova la Calculadora</router-link>
11         </div>
12         <div class="feature-card">
13           <h2>Comentaris</h2>
14           <p>Consulta els comentaris d'usuari i comparteix les teves opinions.</p>
15           <router-link to="/comments" class="feature-link">Veure Comentaris</router-link>
16         </div>
17         <div class="feature-card">
18           <h2>Usuaris</h2>
19           <p>Explora la llista d'usuaris amb informació detallada.</p>
20           <router-link to="/users" class="feature-link">Veure Usuaris</router-link>
21         </div>
22       </div>
23     </div>
24   </div>
25 </template>
```

CalculatorPage.vue

En la pàgina de la calculadora al template, he afegit un div contenidor per a la calculadora, dins un display on es mostrara l'operació, i un conjunt de botons en els quals podrem operar la calculadora.

Son els botons més clàssics en els que he pogut pensar, un clear per a netejar tot el display, clear-entry que esborrara l'últim digit afegit, els parentesis, els simbols per a elevar a 2, elevar a n, l'arrel, els números, suma, multiplicació, divisió i resta, la coma, el % per a calcular el residu i per finalitzar l'igual.

```
▼ CalculatorPage.vue ×
src > views > ▼ CalculatorPage.vue > {} style scoped > .calculator-container
1  <template>
2    <div class="calculator-container">
3      <div class="calculator">
4        <h1 class="title">Calculadora</h1> <!-- El títol dentro del contenidor -->
5        <div class="display">{{ display }}</div>
6        <div class="buttons">
7          <button @click="clearDisplay" class="clear">C</button>
8          <button @click="clearEntry" class="clear-entry">CE</button>
9          <button @click="appendParenthesis('(')"> ( </button>
10         <button @click="appendParenthesis(''))"> ) </button>
11         <button @click="square()" class="operator">x2</button>
12         <button @click="choosePower()" class="operator">^</button>
13         <button @click="squareRoot()" class="operator">√</button>
14         <button @click="operate('x')" class="operator">x</button>
15         <button @click="appendNumber(7)" class="number">7</button>
16         <button @click="appendNumber(8)" class="number">8</button>
17         <button @click="appendNumber(9)" class="number">9</button>
18         <button @click="operate('/')" class="operator">÷</button>
19         <button @click="appendNumber(4)" class="number">4</button>
20         <button @click="appendNumber(5)" class="number">5</button>
21         <button @click="appendNumber(6)" class="number">6</button>
22         <button @click="operate('+')" class="operator">+</button>
23         <button @click="appendNumber(1)" class="number">1</button>
24         <button @click="appendNumber(2)" class="number">2</button>
25         <button @click="appendNumber(3)" class="number">3</button>
26         <button @click="operate('-')" class="operator">-</button>
27         <button @click="operate('%')" class="operator">%</button>
28         <button @click="appendNumber(0)" class="number zero">0</button>
29         <button @click="appendDecimal()" class="number">,</button>
30         <button @click="calculate()" class="equal">=</button>
31       </div>
32     </div>
33   </div>
34 </template>
```

En el script hem afegir la funcionalitat de la calculadora, com data que mostrara la per la pantalla l'operacio i els resultats. En methods guardarem les diferents operacions disponibles en la nostra calculadora.

```
<script>
export default {
  data() {
    return {
      display: '',          // Mostrar en pantalla
      operatorSet: false, // Controla si ya se ingresó un operador
    };
  },
  methods: {
    clearDisplay() {
      this.display = '';
      this.operatorSet = false;
    },
    clearEntry() {
      // Elimina el último valor ingresado
      this.display = this.display.slice(0, -1);
    },
    appendNumber(number) {
      this.display += number.toString();
      this.operatorSet = false;
    },
    appendDecimal() {
      if (!this.display.includes('.')) {
        this.display += '.';
      }
    },
    operate(operator) {
      if (!this.operatorSet && this.display !== '') {
        this.display += operator;
        this.operatorSet = true;
      }
    },
    square() {
      if (this.display) {
        const result = Math.pow(parseFloat(this.display), 2);
        this.display = result.toString();
      }
    },
  },
}
```

En calculate substituir-em certs simbols que utilitzem en la calculadora pero que ficant-los en operadors de javascript ja els podrem utilitzar.

```
choosePower() {
    this.display += '^';
    this.operatorSet = true;
},
squareRoot() {
    if (this.display) {
        const result = Math.sqrt(parseFloat(this.display));
        this.display = result.toString();
    }
},
appendParenthesis(parenthesis) {
    this.display += parenthesis;
},
calculate() {
    try {
        // Reemplaza "^" por "**", "X" por "*", y "÷" por "/"
        let expression = this.display.replace(/\^/g, '**')
            .replace(/x/g, '*')
            .replace(/\//g, '/');
        // Evalúa la expresión
        this.display = eval(expression).toString();
    } catch (e) {
        this.display = 'Error';
    }
},
};
</script>
```

CommentsPage.vue

En la pàgina fetch haurem de fer el fetch dels comentaris, en el meu cas he escollit agafar les següents dades de cada comentari, **nom del comentari**, **comentari**, **email** del que ha comentat.

```
▼ CommentsPage.vue ×
src > views > ▼ CommentsPage.vue > {} style scoped > .comment-card:hover
1  <template>
2    <div class="comments-page">
3      <h1 class="title">User Comments</h1>
4      <div v-if="loading" class="loading">Loading comments...</div>
5      <div v-else class="comments-container">
6        <div v-for="comment in comments" :key="comment.id" class="comment-card">
7          <h3 class="comment-name">{{ comment.name }}</h3>
8          <p class="comment-body">{{ comment.body }}</p>
9          <p class="comment-email">- {{ comment.email }}</p>
10         </div>
11       </div>
12     </div>
13   </template>
14
15 <script>
16 export default {
17   data() {
18     return {
19       comments: [],
20       loading: true, // Estado de carga inicial
21     };
22   },
23   async created() {
24     await this.fetchComments(); // Llama al método para cargar comentarios
25   },
26   methods: {
27     async fetchComments() {
28       try {
29         const response = await fetch("https://jsonplaceholder.typicode.com/comments");
30         if (!response.ok) throw new Error("Error fetching comments");
31         this.comments = await response.json();
32       } catch (error) {
33         console.error("Error al cargar los comentarios:", error);
34       } finally {
35         this.loading = false; // Cambia el estado de carga al finalizar
36       }
37     },
38   },
39 }
40 </script>
```

UsersPage.vue

En la pàgina d'usuaris he utilitzat axis per a mostrar la informació dels usuaris, en el meu cas he mostrat el **nom del usuari, el email i l'edat**

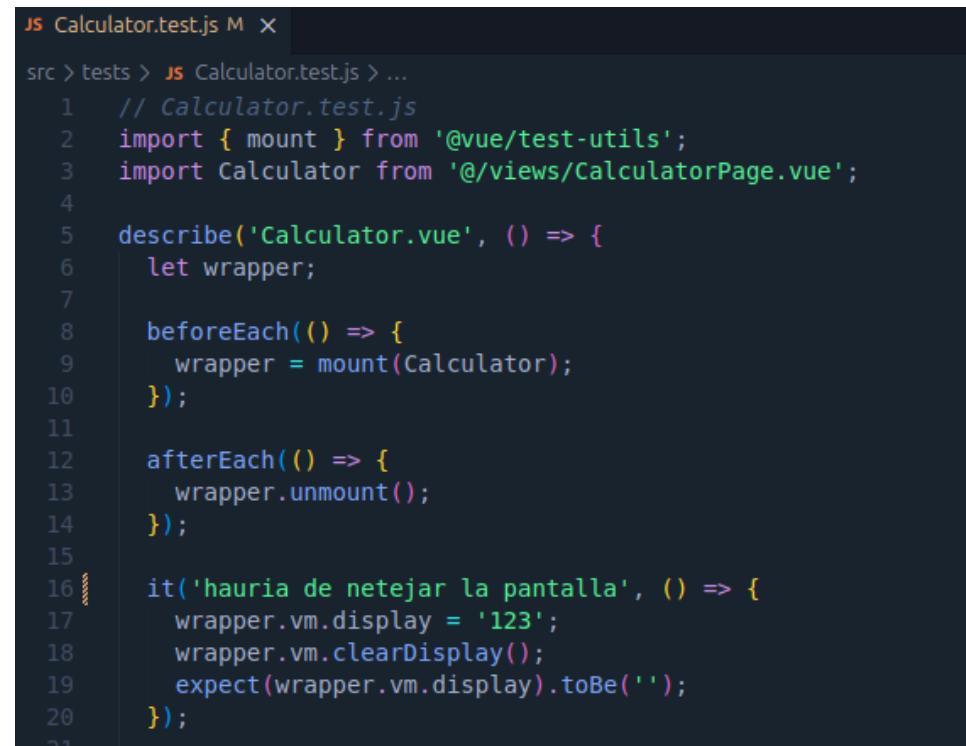
```
▼ UsersPage.vue X
src > views > ▼ UsersPage.vue > {} style scoped
1  <template>
2    <div class="users-page">
3      <h1 class="title">Usuarios</h1>
4      <div v-if="loading" class="loading">Cargando usuarios...</div>
5      <div v-else class="users-container">
6        <div v-for="user in users" :key="user.id" class="user-card">
7          <h3 class="user-name">{{ user.firstName }} {{ user.lastName }}</h3>
8          <p class="user-email">Email: {{ user.email }}</p>
9          <p class="user-age">Edad: {{ user.age }}</p>
10         </div>
11       </div>
12     </div>
13   </template>
14
15 <script>
16 import axios from 'axios';
17
18 export default {
19   name: 'Users',
20   data() {
21     return {
22       users: [],
23       loading: true,
24     };
25   },
26   async created() {
27     try {
28       const response = await axios.get("https://dummyjson.com/users");
29       this.users = response.data.users; // Asegúrate de que `users` es la propiedad correcta
30     } catch (error) {
31       console.error("Error al cargar los usuarios:", error);
32     } finally {
33       this.loading = false;
34     }
35   },
36 };
37 </script>
```

Tests

Calculator.test.js

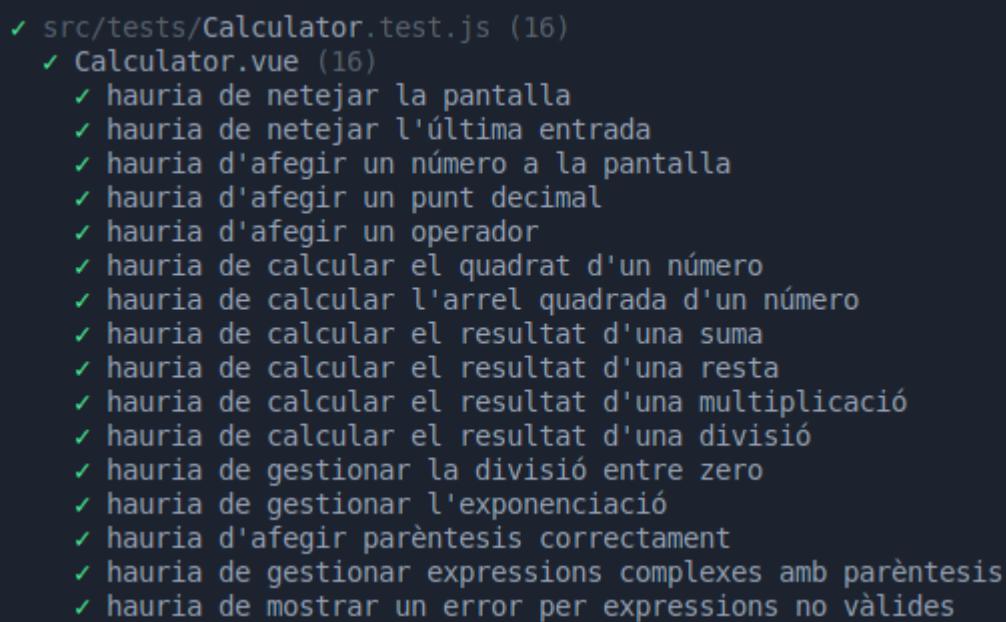
Contingut

En el arxiu de test simplement afegir-em uns tests simples que provin les funcionalitats de la nostra impressora.



```
src > tests > JS Calculator.test.js M X
src > tests > JS Calculator.test.js > ...
1 // Calculator.test.js
2 import { mount } from '@vue/test-utils';
3 import Calculator from '@/views/CalculatorPage.vue';
4
5 describe('Calculator.vue', () => {
6   let wrapper;
7
8   beforeEach(() => {
9     wrapper = mount(Calculator);
10  });
11
12  afterEach(() => {
13    wrapper.unmount();
14  });
15
16  it('hauria de netejar la pantalla', () => {
17    wrapper.vm.display = '123';
18    wrapper.vm.clearDisplay();
19    expect(wrapper.vm.display).toBe('');
20  });
21
```

Prova



```
✓ src/tests/Calculator.test.js (16)
  ✓ Calculator.vue (16)
    ✓ hauria de netejar la pantalla
    ✓ hauria de netejar l'última entrada
    ✓ hauria d'afegir un número a la pantalla
    ✓ hauria d'afegir un punt decimal
    ✓ hauria d'afegir un operador
    ✓ hauria de calcular el quadrat d'un número
    ✓ hauria de calcular l'arrel quadrada d'un número
    ✓ hauria de calcular el resultat d'una suma
    ✓ hauria de calcular el resultat d'una resta
    ✓ hauria de calcular el resultat d'una multiplicació
    ✓ hauria de calcular el resultat d'una divisió
    ✓ hauria de gestionar la divisió entre zero
    ✓ hauria de gestionar l'exponenciació
    ✓ hauria d'afegir parèntesis correctament
    ✓ hauria de gestionar expressions complexes amb parèntesis
    ✓ hauria de mostrar un error per expressions no vàlides
```

Comments.test.js

Contingut

En el test de comentaris farem un petit test que comprovi la petició al fetch si la fa bé i que renderitzi els comentaris després de fer el fetch

```
js Comments.test.js M X
src > tests > js Comments.test.js > describe('CommentsPage.vue') callback > beforeEach() callback
1  import { describe, it, beforeEach, afterEach, expect, vi } from 'vitest';
2  import { shallowMount } from '@vue/test-utils';
3  import CommentsPage from '@/views/CommentsPage.vue';
4
5  describe('CommentsPage.vue', () => {
6    let wrapper;
7
8    beforeEach(async () => {
9      // Estableix un mock global per fetch
10     global.fetch = vi.fn(() =>
11       Promise.resolve({
12         ok: true,
13         json: () => Promise.resolve([
14           {
15             id: 1,
16             name: 'John Doe',
17             body: 'This is a comment.',
18             email: 'john.doe@example.com',
19           },
20           {
21             id: 2,
22             name: 'Jane Doe',
23             body: 'This is another comment.',
24             email: 'jane.doe@example.com',
25           }
26         ])
27       })
28     );
29   });
30
31   it('should render comments after fetch', () => {
32     const comments = wrapper.findAll('.comment');
33     expect(comments.length).toBe(2);
34     expect(wrapper.text()).toContain('John Doe');
35     expect(wrapper.text()).toContain('Jane Doe');
36   });
37
38   it('should handle errors from fetch', () => {
39     global.fetch = vi.fn(() =>
40       Promise.reject(new Error('Network error'))
41     );
42
43     wrapper.update();
44     const errorElement = wrapper.find('.error');
45     expect(errorElement.exists()).toBe(true);
46     expect(errorElement.text()).toContain('Network error');
47   });
48
49   it('should handle multiple errors from fetch', () => {
50     global.fetch = vi.fn(() =>
51       Promise.reject(new Error('Network error'))
52     );
53
54     wrapper.update();
55     const errorElement = wrapper.find('.error');
56     expect(errorElement.exists()).toBe(true);
57     expect(errorElement.text()).toContain('Network error');
58   });
59
60   it('should handle multiple errors from fetch', () => {
61     global.fetch = vi.fn(() =>
62       Promise.reject(new Error('Network error'))
63     );
64
65     wrapper.update();
66     const errorElement = wrapper.find('.error');
67     expect(errorElement.exists()).toBe(true);
68     expect(errorElement.text()).toContain('Network error');
69   });
70
71   it('should handle multiple errors from fetch', () => {
72     global.fetch = vi.fn(() =>
73       Promise.reject(new Error('Network error'))
74     );
75
76     wrapper.update();
77     const errorElement = wrapper.find('.error');
78     expect(errorElement.exists()).toBe(true);
79     expect(errorElement.text()).toContain('Network error');
80   });
81
82   it('should handle multiple errors from fetch', () => {
83     global.fetch = vi.fn(() =>
84       Promise.reject(new Error('Network error'))
85     );
86
87     wrapper.update();
88     const errorElement = wrapper.find('.error');
89     expect(errorElement.exists()).toBe(true);
90     expect(errorElement.text()).toContain('Network error');
91   });
92
93   it('should handle multiple errors from fetch', () => {
94     global.fetch = vi.fn(() =>
95       Promise.reject(new Error('Network error'))
96     );
97
98     wrapper.update();
99     const errorElement = wrapper.find('.error');
100    expect(errorElement.exists()).toBe(true);
101    expect(errorElement.text()).toContain('Network error');
102  });
103});
```

Prova

```
✓ src/tests/Comments.test.js (1)
  ✓ CommentsPage.vue (1)
    ✓ hauria de renderitzar els comentaris després de fer el fetch

Test Files 1 passed (1)
  Tests 1 passed (1)
  Start at 17:03:32
  Duration 127ms
```

Users.test.js

Contingut

En el test d'usuaris el mateix, que carregui i renderitzi correctament la petició axios.

```
js Users.test.js M ×

src > tests > js Users.test.js > ⚡ describe('UsersPage.vue') callback > ⚡ it('hauria de renderitzar els la peticio dels usuaris') callback
  1 import { describe, it, beforeEach, afterEach, expect, vi } from 'vitest';
  2 import { shallowMount } from '@vue/test-utils';
  3 import axios from 'axios';
  4 import UsersPage from '@/views/UsersPage.vue';
  5
  6 // Simula el mòdul de axios
  7 vi.mock('axios');
  8
  9 describe('UsersPage.vue', () => {
10   let wrapper;
11
12   beforeEach(async () => {
13     // Configura el mock de axios amb la resposta desitjada
14     axios.get.mockResolvedValue({
15       data: {
16         users: [
17           { id: 1, firstName: 'John', lastName: 'Doe', email: 'john.doe@example.com', age: 30 },
18           { id: 2, firstName: 'Jane', lastName: 'Doe', email: 'jane.doe@example.com', age: 28 },
19         ],
20       },
21     });
22   });
23
24   it('hauria de renderitzar els la peticio dels usuaris', () => {
25     const wrapper = shallowMount(UsersPage);
26
27     expect(wrapper.html()).toContain('John Doe');
28     expect(wrapper.html()).toContain('Jane Doe');
29   });
30
31   // Add more tests here...
32
33 });
34
35 
```

Prova

✓ src/tests/Users.test.js (1)
 ✓ UsersPage.vue (1)
 - hauria de renderitzar els la noticia dels usuaris

```
Test Files  1 passed (1)
          Tests  1 passed (1)
          Start at 17:05:56
          Duration 161ms
```

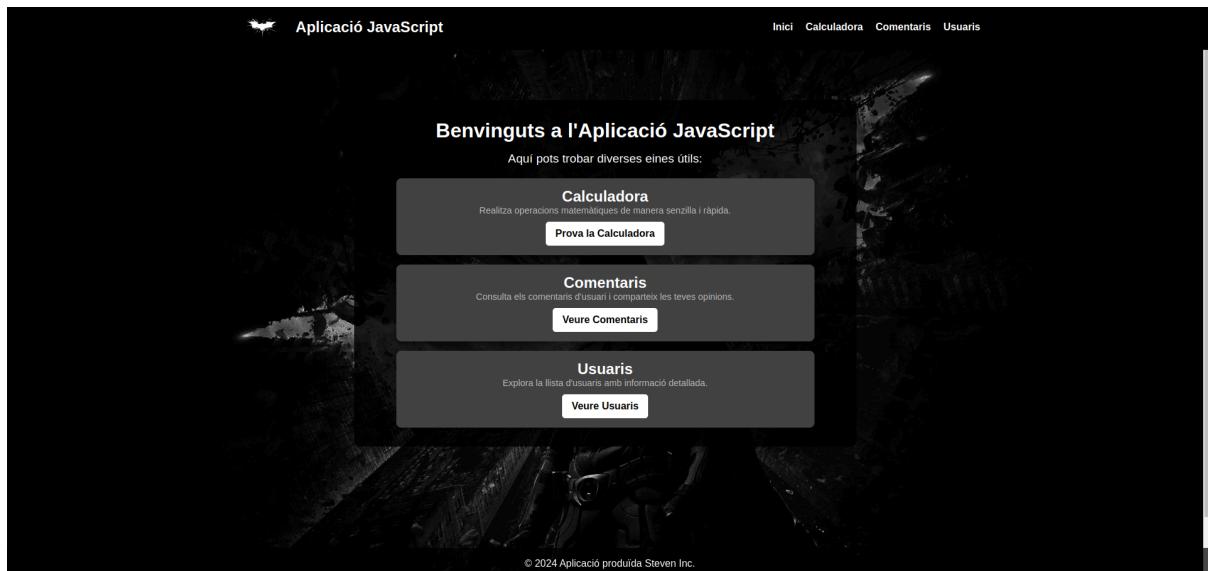
Resultat

Header Responsiu



LandingPage

Pantalla completa

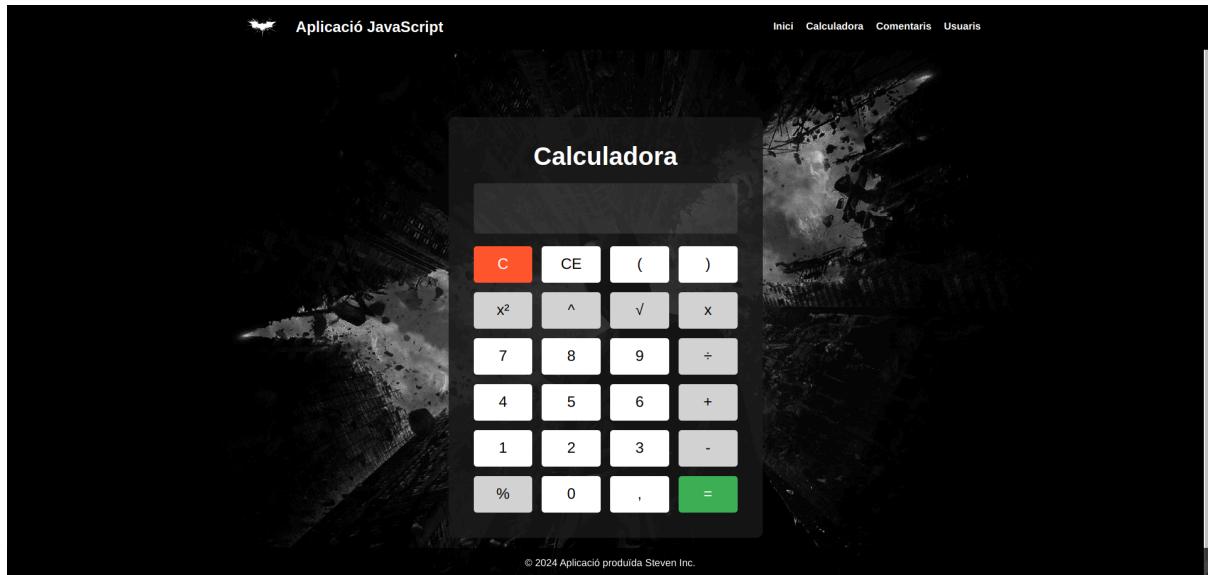


Responsive

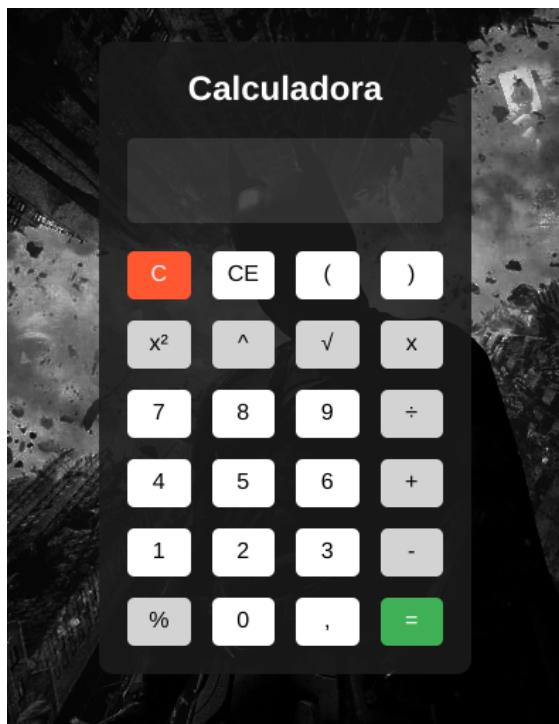


CalculatorPage

Pantalla completa



Responsive



CommentsPage

Pantalla completa

The screenshot shows a web application titled "Aplicació JavaScript" with a navigation bar including "Inici", "Calculadora", "Comentaris", and "Usuaris". The main content area is titled "User Comments" and displays five comments in dark gray boxes:

- id labore ex et quam laborum**
laudantium enim quasi est quidem magnam voluptate ipsam eos tempora quo necessitatibus dolor quam autem quasi reiciendis et nam sapiente accusantium
- Eliseo@gardner.biz
- quo vero reiciendis velit similique earum**
est natus enim nihil est dolore omnis voluptatem numquam et omnis occaecati quod ullam at voluptatem error expedita pariatur nihil sint nostrum voluptatem reiciendis et
- Jayne_Kuhic@sydney.com
- odio adipisci rerum aut animi**
quia molestiae reprehenderit quasi aspernatur aut expedita occaecati aliquam eveniet laudantium omnis quibusdam delectus saepe quia accusamus maiores nam est cum et ducimus et vero voluptates excepturi defeniti ratione
- Nikita@garfield.biz
- alias odio sit**
non et atque occaecati deserunt quas accusantium unde odit nobis qui voluptatem quia voluptas consequuntur itaque dolor et qui rerum defeniti ut occaecati
- Lew@alysha.tv
- vero eaque aliquid doloribus et culpa**
Habemus modis et ratione consequuntur itaque dolor et qui rerum defeniti ut occaecati
© 2024 Aplicació produïda Steven Inc.

Responsive

The screenshot shows the same "User Comments" section from the previous image, but it is displayed on a smaller screen, likely a mobile phone. The content is scaled down and wrapped to fit the narrower width, while maintaining the overall layout and styling of the original application.

UsersPage

Pantalla Completa

The screenshot shows a web application titled "Aplicació JavaScript" with a dark background featuring a city skyline at night. At the top right, there are navigation links: "Inici", "Calculadora", "Comentaris", and "Usuaris". The main title "Usuarios" is centered above a list of user profiles. Each profile card contains the user's name, email, and age. The profiles listed are:

- Emily Johnson**
Email: emily.johnson@x.dummyjson.com
Edad: 28
- Michael Williams**
Email: michael.williams@x.dummyjson.com
Edad: 35
- Sophia Brown**
Email: sophia.brown@x.dummyjson.com
Edad: 42
- James Davis**
Email: james.davis@x.dummyjson.com
Edad: 45
- Emma Miller**
Email: emma.miller@x.dummyjson.com
Edad: 30
- Olivia Wilson**
Email: olivia.wilson@x.dummyjson.com
Edad: 22
- Alexander Jones**
Email: alexander.jones@x.dummyjson.com
Edad: 38

At the bottom right of the page, there is a copyright notice: "© 2024 Aplicació produïda Steven Inc."

Responsive

The screenshot shows the same "Usuarios" page but with a single user profile expanded. The user "Emily Johnson" is highlighted with a larger card. Her details are as follows:

Emily Johnson
Email: emily.johnson@x.dummyjson.com
Edad: 28

The other five user profiles are shown in smaller, semi-transparent cards below it.

Github

<https://github.com/ManuelSteven21/projecteJavaScript>

Netify

<https://672a42d4039a4200084e31e2--meek-llama-40a77d.netlify.app/>