

Projecte JAVASCRIPT



Professor: Jordi Vega

Assignatura: M7 Desenvolupament d'interfícies

Nom: Mihai Sebastian

Cognoms: Burghilea

INDEX

Que s'ha de fer?	3
Que s'ha d'entregar?	4
Data d'entrega	4
Creació del projecte	5
Creació de les pàgines:	7
ERROR 404:	7
LandingPage	8
Calculadora	9
Api utilitzant Fetch i API Axios	10
Instal·lació Tailwind	12
HEADER I FOOTER	13
Landing Page:	16
Calculadora	17
API FETCH	21
API AXIOS	22
Resultat	23
TESTS	25
Render	29



Que s'ha de fer?

Una aplicació web utilitzant Javascript amb el framework Vue tipus SPA. S'ha de crear un nou projecte de vue per fer l'aplicació.

Descripció de les pàgines:

- A la pàgina principal una landing page tipus portada/presentació del projecte.
- A la segona pàgina ha d'haver una calculadora amb diversos botons per als números i les operacions (suma, resta, multiplicació, divisió, etc), s'ha de contemplar els decimals en les operacions, també s'ha de poder esborrar el contingut de la pantalla.
- A la tercera pàgina s'ha d'utilitzar fetch per poder mostrar les dades de la api de <https://jsonplaceholder.typicode.com/comments>
- A la quarta pàgina s'ha d'utilitzar axios per poder mostrar les dades de la api de <https://dummyjson.com/users>

Descripció dels tests:

- El primer test ha de comprovar tota la funcionalitat de la calculadora
- El segon test ha de comprovar tot el funcionament de la petició en fetch
- El tercer test ha de comprovar tot el funcionament de la petició en axios

Coses extra que ha de tindre l'aplicació:

- Header/navbar per a navegar entre les diferents pàgines.
- Footer per mostrar la informació del creador de la pàgina.
- Ha de ser responsive.
- S'ha de poder navegar entre les pàgines en la seva ruta corresponent
- S'ha de reaprofitar el codi sempre que sigui possible.

La documentació ha de tindre:

- Portada
- Índex
- Captura i explicació dels fitxers que heu afegit i modificat, i perquè
- Comprovacions del funcionament

El codi s'ha de pujar a un servidor a triar una opció entre:

- Netlify
- OnRender



Que s'ha d'entregar?

- A github
 - Codi
- Al servidor escollit (Netlify/OnRender)
 - Heu de pujar el vostre codi per pujar-ho a explotació
- Al moodle
 - Documentació i enllaç al codi del github i explotació

Data d'entrega

Dimarts 5/11/2024

Creació del projecte

Creo el projecte activant l'opció de VUE Router per a la creació de un projecte SPA.

```
alumnat@sebastian:~/Code$ npm create vue@latest

> npx
> create-vue

Vue.js - The Progressive JavaScript Framework

✓ Project name: ... ProjecteVUE
✓ Package name: ... projectevue
✓ Add TypeScript? ... No / Yes
✓ Add JSX Support? ... No / Yes
✓ Add Vue Router for Single Page Application development? ... No / Yes
✓ Add Pinia for state management? ... No / Yes
✓ Add Vitest for Unit Testing? ... No / Yes
✓ Add an End-to-End Testing Solution? > No
✓ Add ESLint for code quality? ... No / Yes
✓ Add Vue DevTools 7 extension for debugging? (experimental) ... No / Yes

Scaffolding project in /home/alumnat/Code/ProjecteVUE...

Done. Now run:

  cd ProjecteVUE
  npm install
  npm run dev
```

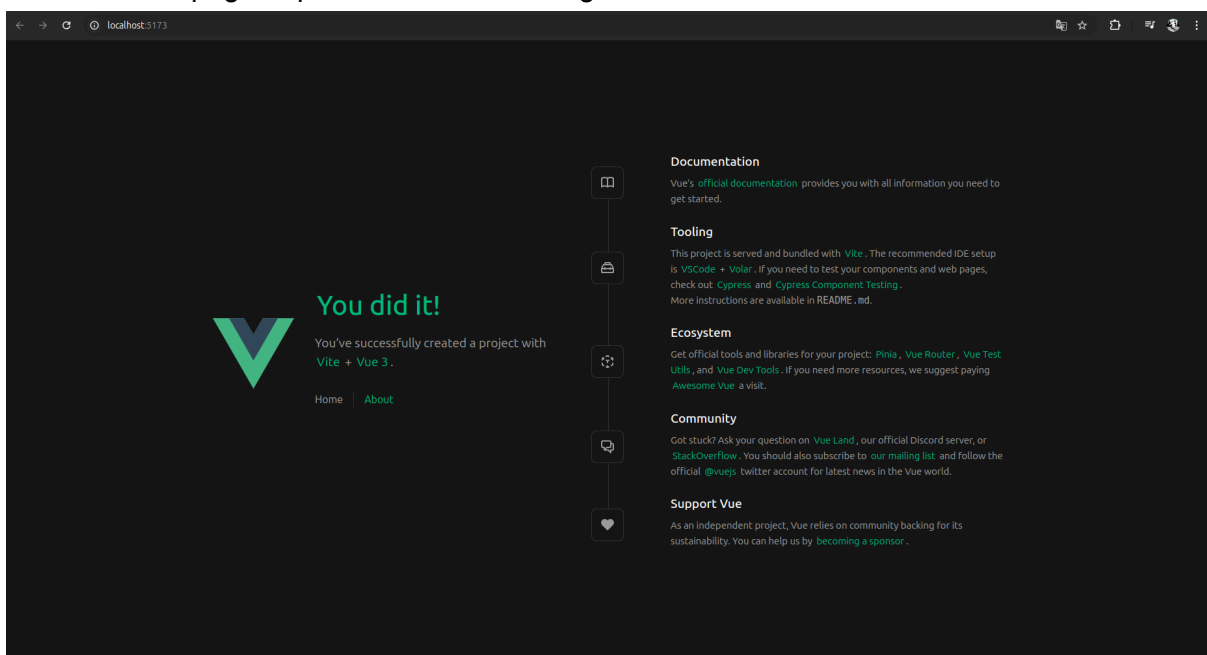
Després faig un npm install per instal·lar les dependències i npm run dev per executar els scripts i poder accedir a la pàgina per web.

```
alumnat@sebastian:~/Code/ProjecteVUE$ npm install
added 32 packages, and audited 33 packages in 13s
5 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
alumnat@sebastian:~/Code/ProjecteVUE$ npm run dev
> projectevue@0.0.0 dev
> vite

VITE v5.4.10  ready in 899 ms

➔ Local:   http://localhost:5173/
➔ Network: use --host to expose
➔ press h + enter to show help
```

Per defecte la pàgina quan entrem surt el següent:



Creació de les pàgines:

ERROR 404:

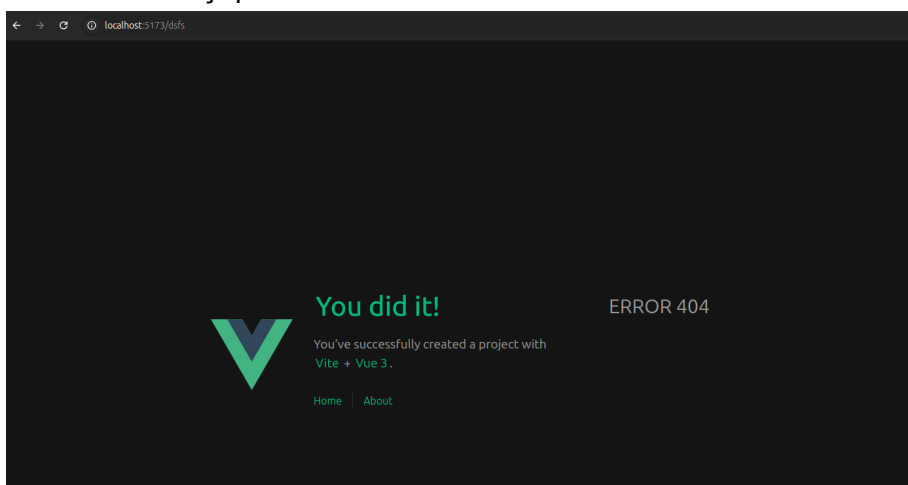
Primer que tot fafe la pàgina error 404 utilitzant pathMatch i cridant a la vista notFound.vue.

```
{  
  path: '/*:pathMatch(.*)*',  
  name: 'notFound',  
  // route level code-splitting  
  // this generates a separate chunk (About.[hash].js) for this route  
  // which is lazy-loaded when the route is visited.  
  component: () => Promise<{...}> => import('../views/notFound.vue')  
}
```

notFound.vue temporal:

```
<template> Show component usages  
  <h1>ERROR 404</h1>  
</template>  
<script>  
  export default { Show usages  
    name: 'NotFound'  
  }  
</script>
```

Al ficar un enllaç que no existeix sortirà error 404.



LandingPage

Elimino tot de APP.VUE i deixo un router-link per navegar entre pàgines:

```

1  <script setup> Show component usages
2  import { RouterLink, RouterView } from 'vue-router'
3  </script>
4
5  <template>
6    <header>
7      <nav>
8        <RouterLink to="/">Home</RouterLink>
9      </nav>
10   </header>
11
12   <RouterView />
13 </template>
14
15 <style scoped>
16 </style>

```

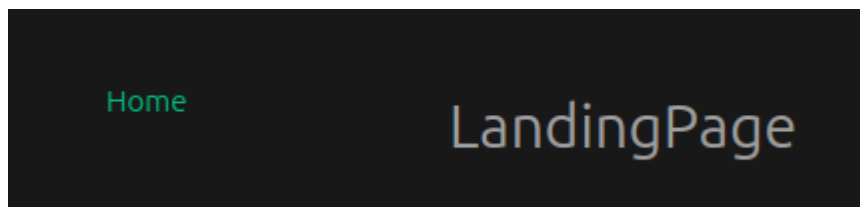
A la view de home igual, ja que serà aquesta la que utilitzaré per a la creació de la landing page

```

1  <script setup> Show component usages
2  <script>
3
4  <template>
5    <h1>LandingPage</h1>
6  </template>
7

```

Queda així de moment:



Calculadora

Faré el mateix amb les altres pàgines:

```
1 <script setup> Show component usages
2
3 </script>
4
5 <template>
6   <h1>Calculadora</h1>
7 </template>
8
9 <style scoped>
10
11 </style>
```

```
{
  path: '/calculadora',
  name: 'Calculadora',
  // route level code-splitting
  // this generates a separate chunk (About.[hash].js) for this route
  // which is lazy-loaded when the route is visited.
  component: () => Promise<{...}> => import('../views/Calculadora.vue')
},
```

```
<template>
  <header>
    <nav>
      <RouterLink to="/">Home</RouterLink>
      <RouterLink to="/Calculadora">Calculadora</RouterLink>
      <RouterLink to="/">Api Fetch</RouterLink>
      <RouterLink to="/">Api Axios</RouterLink>
    </nav>
  </header>

  <RouterView />
</template>
```

Home Calculadora Api Fetch Api Axios **Calculadora**

Api utilitzant Fetch i API Axios

```
{
  path: '/api-fetch',
  name: 'ApiFetch',
  // route level code-splitting
  // this generates a separate chunk (About.[hash].js) for this route
  // which is lazy-loaded when the route is visited.
  component: () :Promise<{...}> => import('../views/APIFetch.vue')
},
{
  path: '/api-axios',
  name: 'ApiAxios',
  // route level code-splitting
  // this generates a separate chunk (About.[hash].js) for this route
  // which is lazy-loaded when the route is visited.
  component: () :Promise<{...}> => import('../views/APIAxios.vue')
```

```
1  <script setup> Show component
2
3  </script>
4
5  <template>
6    <h1>Api AXIOS</h1>
7  </template>
8
9  <style scoped>
10    ⚡
11  </style>
```

```
1  <script setup> Show component
2
3  </script>
4
5  <template>
6    <h1>Api Fetch</h1>
7  </template>
8
9  <style scoped>
10
11  </style>
```

```
<template>
  <header>
    <nav>
      <RouterLink to="/">Home</RouterLink>
      <RouterLink to="/Calculadora">Calculadora</RouterLink>
      <RouterLink to="/api-fetch">Api Fetch</RouterLink>
      <RouterLink to="/api-axios">Api Axios</RouterLink>
    </nav>
  </header>

  <RouterView />
</template>
```

Home Calculadora Api Fetch Api Axios

Api Fetch

Home Calculadora Api Fetch Api Axios

Api AXIOS

Instal·lació Tailwind

Per a la instal·lació de tailwind faré un npm install i npx tailwindcss init -p per crear els arxius.

```
alumnat@sebastian:~/Code/ProjecteVUE$ npm install -D tailwindcss@latest postcss@latest autoprefixer@latest

added 118 packages, and audited 151 packages in 3s

37 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
alumnat@sebastian:~/Code/ProjecteVUE$ npx tailwindcss init -p

Created Tailwind CSS config file: tailwind.config.js
Created PostCSS config file: postcss.config.js
```

Després modificaré el content per indicar a on vull que s'apliqui el tailwind.

```
1  /** @type {import('tailwindcss').Config} */
2  export default {
3    content: ['./index.html', './src/**/*.vue,js,ts,jsx,tsx'],
4    theme: {
5      extend: {},
6    },
7    plugins: [],
8  }
```

El contingut de postcss el deixaré amb la modificació per defecte:

```
1  export default {
2    plugins: [
3      tailwindcss,
4      autoprefixer,
5    ],
6  }
```

Per últim creo el fitxer index.css i afegeixo els components i utilitats de tailwind.

```
1  /* ./src/index.css */
2  @tailwind base;
3  @tailwind components;
4  @tailwind utilities;
```

A main.js elimino main.css i importo index.css per utilitzar vue amb tailwind.

```
1
2 import { createApp } from 'vue'
3 import App from './App.vue'
4 import './index.css'
5
6 import router from './router'
7
8 const app : App<Element> = createApp(App)
9
10 app.use(router)
11
12 app.mount( { rootContainer: '#app' } )
13
```

HEADER I FOOTER

Primer instal·lo headlessui i heroicons amb npm

```
alumnat@sebastian:~/Code/ProjecteVUE$ npm install @headlessui/vue
added 3 packages, and audited 154 packages in 2s

39 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
alumnat@sebastian:~/Code/ProjecteVUE$ npm install @heroicons/vue
added 1 package, and audited 155 packages in 972ms

39 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```

1  <template> Show component usages
2  <header class="bg-gray-900">
3    <nav class="mx-auto flex max-w-7xl items-center justify-between p-6 lg:px-8" aria-label="Global">
4      <div class="flex lg:flex-1">
5        <a href="#" class="-m-1.5 p-1.5">
6          <span class="sr-only">Your Company</span>
7          
8        </a>
9      </div>
10     <div class="flex lg:hidden">
11       <button type="button" class="-m-2.5 inline-flex items-center justify-center rounded-md p-2.5 text-gray-400" @click="mobileMenuOpen = true">
12         <span class="sr-only">Open main menu</span>
13         <Bars3Icon class="h-6 w-6" aria-hidden="true" />
14       </button>
15     </div>
16     <div class="hidden lg:flex lg:gap-x-12">
17       <a v-for="item in navigation" :key="item.name" :href="item.href" class="text-sm/6 font-semibold text-white">{{ item.name }}</a>
18     </div>
19     <div class="hidden lg:flex lg:flex-1 lg:justify-end">
20       <a href="#" class="text-sm/6 font-semibold text-white">Log in <span aria-hidden="true">&srarr;</span></a>
21     </div>
22   </nav>
23   <Dialog class="lg:hidden" @close="mobileMenuOpen = false" :open="mobileMenuOpen">
24     <div class="fixed inset-0 z-10" />
25     <DialogPanel class="fixed inset-y-0 right-0 z-10 w-full overflow-y-auto bg-gray-900 px-6 py-6 sm:max-w-sm sm:ring-1 sm:ring-white/10">
26       <div class="flex items-center justify-between">
27         <a href="#" class="-m-1.5 p-1.5">
28           <span class="sr-only">Your Company</span>
29           
30         </a>

```

```

1 <template> Show component usages
2 <footer class="bg-gray-900">
3   <div class="mx-auto max-w-7xl px-6 py-8 md:flex md:items-center md:justify-between lg:px-8">
4     <div class="flex justify-center gap-x-6 md:order-2">
5       <a href="#">
6         <img alt="Facebook icon" data-bbox="100 880 120 900" class="h-8 w-8" />
7       </a>
8     </div>
9     <p class="mt-8 text-center text-sm/6 text-gray-400 md:order-1 md:mt-0">© 2024 Your Company, Inc. All rights reserved.</p>
10   </div>
11 </footer>
12 </template>
13
14 <script setup>
15 import { defineComponent, h } from 'vue'
16
17 const navigation = [
18   {
19     name: 'Facebook',
20     href: '#',
21     icon: defineComponent( options: {
22       render: () =>
23         h( type: 'svg', props: { fill: 'currentColor', viewBox: '0 0 24 24' }, children: [
24           h( type: 'path', props: {
25             'fill-rule': 'evenodd',
26             d: 'M22 12c0-5.523-4.477-10-10-10S2 6.477 2 12c0 4.991 3.657 9.128 8.438 9.878v-6.987h-2.54V12h2.54V9.797c0-2.506 1.492-3.89 3.777-3.89 1.094 0 2.188 0.406 3.282 1.103
27             'clip-rule': 'evenodd',
28           })
29         ]),
30     }),
31   },
32 ],
33 {
34   name: 'Instagram',

```

A app.vue importo els components i faig la crida al template, també afegeixo un poc de css per a que el footer estigui baix sempre.

```
1  <script setup> Show component usages
2  import { RouterLink, RouterView } from 'vue-router'
3  import Header from './components/Header.vue';
4  import Footer from './components/Footer.vue';
5  </script>
6
7  <template>
8    <div class="flex min-h-screen flex-col">
9      <Header />
10
11     <main class="flex-grow">
12       <RouterView />
13     </main>
14
15     <Footer />
16   </div>
17 </template>
18
19 <style scoped>
20 </style>
```

Landing Page:

Contingut amb tailwind de la landing page:

```

1 <template> Show component usages
2 <div class="bg-white py-24 sm:py-32">
3   <div class="mx-auto max-w-7xl px-6 lg:px-8">
4     <div class="mx-auto grid max-w-2xl grid-cols-1 items-start gap-x-8 gap-y-16 sm:gap-y-24 lg:mx-0 lg:max-w-none lg:grid-cols-2">
5       <div class="lg:pr-4">
6         <div class="relative overflow-hidden rounded-3xl bg-gray-900 px-6 pb-9 pt-64 shadow-2xl sm:px-12 lg:max-w-lg lg:px-8 lg:pb-8 xl:px-12">
7           
8           <div class="absolute inset-0 bg-gray-900 mix-blend-multiply" />
9           <figure class="relative isolate">
10            
11            <blockquote class="mt-6 text-xl/8 font-semibold text-white">
12              <p>"Descobreix l'aplicació web SPA creada amb Vue, que integra funcionalitats com una calculadora completa, consultes d'API amb JSONPlaceholder, i una interfície d'usuari moderna i intuïtiva."</p>
13            </blockquote>
14            <figcaption class="mt-6 text-sm/6 text-gray-300">
15              <strong class="font-semibold text-white">MihaiSebastian,</strong> Creador de l'Aplicació
16            </figcaption>
17          </figure>
18        </div>
19      </div>
20      <div>
21        <div class="text-base/7 text-gray-700 lg:max-w-lg">
22          <p class="text-base/7 font-semibold text-indigo-600">Descripció del Projecte</p>
23          <h1 class="mt-2 text-pretty text-4xl font-semibold tracking-tight text-gray-900 sm:text-5xl">
24            Aplicació SPA amb Vue per a una Experiència Completa
25          </h1>
26          <div class="max-w-xl">
27            <p class="mt-6">Aquest projecte és una aplicació web tipus SPA creada amb Vue, que inclou diverses pàgines i funcionalitats per a una experiència d'usuari completa i intuïtiva.
28            <p class="mt-8">La primera pàgina és una landing page que serveix com a presentació del projecte. La segona pàgina ofereix una interfície d'usuari moderna i intuïtiva.
29            <p class="mt-8">La tercera pàgina utilitza fetch per consultar dades des de l'API de JSONPlaceholder, i la quarta pàgina utilitza axios per fer consultes d'API amb JSONPlaceholder.
30          </div>
31        </div>
32        <dl class="mt-10 grid grid-cols-2 gap-8 border-t border-gray-900/10 pt-10 sm:grid-cols-4">
33          <div v-for="(stat, statIdx) in stats" :key="statIdx">

```


Calculadora

Per a la calculadora creo un span amb `{{displayValue}}` que serà el valor de les operacions que es mostraran en pantalla.

Als botons ficaré un `@click` per a que quan detecta el click del usuari crida a una funció javascript.

```

1 <template> Show component usages
2 <div class="flex items-start justify-center pt-10 bg-cyan-900">
3   <div class="w-full max-w-md bg-gray-900 shadow-lg rounded-lg overflow-hidden">
4     <div class="p-4">
5       <!-- Display -->
6       <div class="text-right bg-gray-800 text-white text-4xl font-mono p-4 rounded-md mb-4">
7         <span>{{ displayValue }}</span>
8       </div>
9
10      <!-- Buttons -->
11      <div class="grid grid-cols-4 gap-2">
12        
13        <button @click="inputOperation( operator: '%')*" class="bg-gray-600 text-white text-2xl p-4 rounded-md">%</button>
14        <button @click="clearDisplay" class="bg-gray-600 text-white text-2xl p-4 rounded-md">C</button>
15        <button @click="deleteDigit" class="bg-gray-600 text-white text-2xl p-4 rounded-md">DEL</button>
16
17        <button @click="inputOperation( operator: '1/x')*" class="bg-gray-600 text-white text-2xl p-4 rounded-md">1/x</button>
18        <button @click="inputOperation( operator: 'x^2')*" class="bg-gray-600 text-white text-2xl p-4 rounded-md">x^2</button>
19        <button @click="inputOperation( operator: 'sqrt')*" class="bg-gray-600 text-white text-2xl p-4 rounded-md">√x</button>
20        <button @click="inputOperation( operator: '/')*" class="bg-gray-600 text-white text-2xl p-4 rounded-md">÷</button>
21
22        <button @click="inputDigit( digit: '7')*" class="bg-green-500 text-white text-2xl p-4 rounded-md">7</button>
23        <button @click="inputDigit( digit: '8')*" class="bg-green-500 text-white text-2xl p-4 rounded-md">8</button>
24        <button @click="inputDigit( digit: '9')*" class="bg-green-500 text-white text-2xl p-4 rounded-md">9</button>
25        <button @click="inputOperation( operator: '*')*" class="bg-gray-600 text-white text-2xl p-4 rounded-md">×</button>
26
27        <button @click="inputDigit( digit: '4')*" class="bg-green-500 text-white text-2xl p-4 rounded-md">4</button>
28        <button @click="inputDigit( digit: '5')*" class="bg-green-500 text-white text-2xl p-4 rounded-md">5</button>
29        <button @click="inputDigit( digit: '6')*" class="bg-green-500 text-white text-2xl p-4 rounded-md">6</button>
30        <button @click="inputOperation( operator: '-')*" class="bg-gray-600 text-white text-2xl p-4 rounded-md">-</button>
31

```

Al script afegiré el codi javascript, primer crearé la operació `displayValue` amb el valor 0 per a mostrar en pantalla quan no hi ha més contingut.

Després començo a programar les funcions, per a la funció decimal faig un `split` pel numerador i si no està el punt l'afegeixo a `displayValue`.

```

<script setup>
import {ref} from 'vue'
const displayValue = ref( value: '0')
let lastOperation = false

function inputDecimal() { Show usages MihaiSebastian *
  const lastNumber = displayValue.value.split(/[\+|-|\*|\/|%]/).pop();
  if (!lastNumber.includes('.')) {
    displayValue.value += '.';
    lastOperation = false;
  }
}

```

Per a la funció clearDisplay fico que el displayValue sigue 0.

Per a la funció de inputDigit el que faré és concatenar el dígit amb display.value.

L'objectiu de lastOperation: és un pivot que el que fa és una vegada té el resultat final (després de fer click al botó igual) retorna un número (el resultat), si es vol continuar amb aquest número com a operació s'ha de fer click a un operador, si es fa click a un número se olvida la memòria de la última operació i se comença una operació nova.

```
function clearDisplay() { Show usages MihaiSebastian
  displayValue.value = '0'
  lastOperation = false
}

function inputDigit(digit) { Show usages MihaiSebastian
  if (lastOperation === true){
    displayValue.value = digit
    lastOperation = false
  }
  else if (displayValue.value === '0') {
    displayValue.value = digit
  }
  else {
    displayValue.value += digit
  }
}
```

Per a la funció `inputOperation` tinc diferents condicionals per saber quin ha triat l'usuari, depen del que sigui executat la fórmula ($1/x$, x^2 , \sqrt{x}) o concatena l'operador (suma, resta, multiplicació, divisió, residu).

També tinc un slice per eliminar l'últim operador i evitar que l'usuari fiqui ($2+*****$).

```
function inputOperation(operator) { Show usages  MihaiSebastian *
  const operators = ['+', '-', '*', '/', '%', '1/x', 'x^2', 'sqrt'];
  const currentValue = parseFloat(displayValue.value);

  if (['+', '-', '*', '/', '%'].includes(operator)) {
    if (operators.includes(displayValue.value.toString().slice(-1))) {
      displayValue.value = displayValue.value.slice(0, -1) + operator;
      lastOperation = false;
    } else {
      displayValue.value += operator;
      lastOperation = false;
    }
  }
  else if (operator === '1/x') {
    displayValue.value = (1 / currentValue).toString();
    lastOperation = true;
  }
  else if (operator === 'x^2') {
    displayValue.value = (currentValue ** 2).toString();
    lastOperation = true;
  }
  else if (operator === 'sqrt') {
    displayValue.value = Math.sqrt(currentValue).toString();
    lastOperation = true;
  }
}
```

Per a la funció calculateResult utilitzo eval per fer el càlcul i fico lastOperation en True, a les altres funcions lastOperation se fica en false.

Per a la funció deleteDigit faig un slice per eliminar l'últim caràcter afegit.

Per a la funció toggleSign retorno la cadena amb el - davant.

```
function calculateResult() { Show usages MihaiSebastian +1
  displayValue.value = eval(displayValue.value)
  displayValue.value = displayValue.value.toString()
  lastOperation = true
}
function deleteDigit() { Show usages MihaiSebastian +1

  displayValue.value = displayValue.value.slice(0, -1)
  if (displayValue.value === '') {
    displayValue.value = '0'
    lastOperation = false
  }
}
function toggleSign() { Show usages Mihai-Sebastian +1 *
  if (displayValue.value === '0') {
    return
  }
  displayValue.value = -parseFloat(displayValue.value)
  lastOperation = false
}
</script>
```

API FETCH

A la view de fetch crido a la api jsonplaceholder/comments.

El que faig es crear la llista comments i retornar el valor de la crida de la api en json.

Després al template faig un for de comments per mostrar el contingut 1 a 1.

```
1  <script setup> Show component usages
2  import {onMounted, ref} from 'vue'
3
4  const comments = ref( value: [])
5  onMounted( hook: () => {
6    fetch( input: 'https://jsonplaceholder.typicode.com/comments')
7      .then(response => response.json())
8      .then(data => {
9        comments.value = data
10      })
11    .catch(error => console.log(error))
12  })
13 </script>
14
15 <template>
16   <div class="pb-4">
17     <div class="max-w-3xl mx-auto mt-8 p-6 bg-gray-100 shadow-md rounded-lg">
18       <h1 class="text-3xl font-bold text-center text-gray-900 mb-6">Comments</h1>
19       <ul class="divide-y divide-black">
20         <li
21           v-for="comment in comments"
22           :key="comment.id"
23           class="py-4 flex flex-col sm:flex-row sm:items-center">
24           <div class="flex-1">
25             <h2 class="text-lg font-semibold text-gray-800">{{ comment.name }}</h2>
26             <p class="text-gray-600 mt-1">{{ comment.body }}</p>
27           </div>
28           <p class="text-gray-500 text-sm mt-2 sm:mt-0 sm:ml-4">{{ comment.email }}</p>
29         </li>
30       </ul>
31     </div>
32   </div>
33 </template>
```

API AXIOS

Per a la API de Axios primer faig un npm install axios.

```
alumnat@sebastian:~/Code/ProjecteVUE$ npm install axios

added 9 packages, and audited 164 packages in 1s

40 packages are looking for funding
  run `npm fund` for details

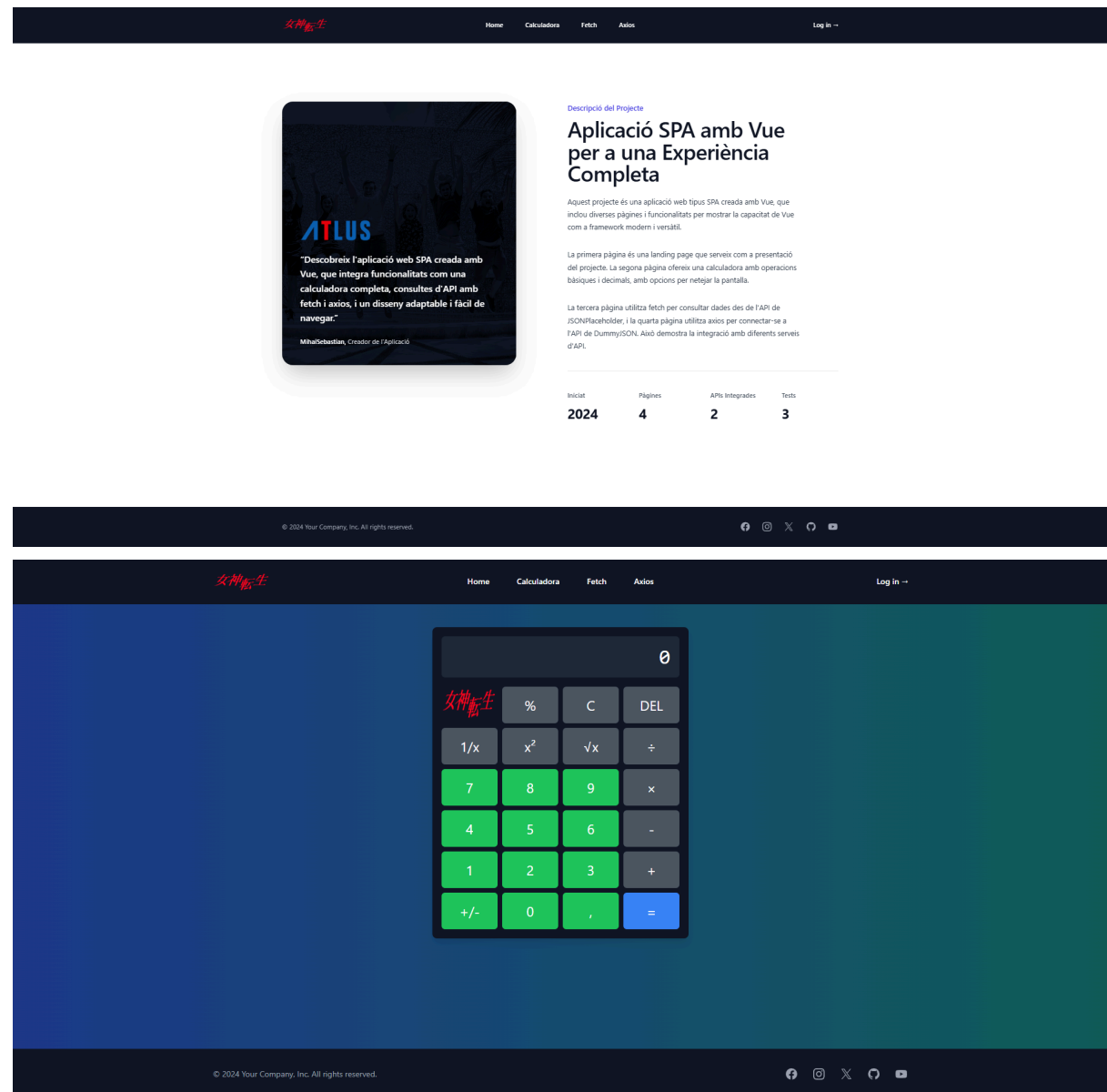
found 0 vulnerabilities
```

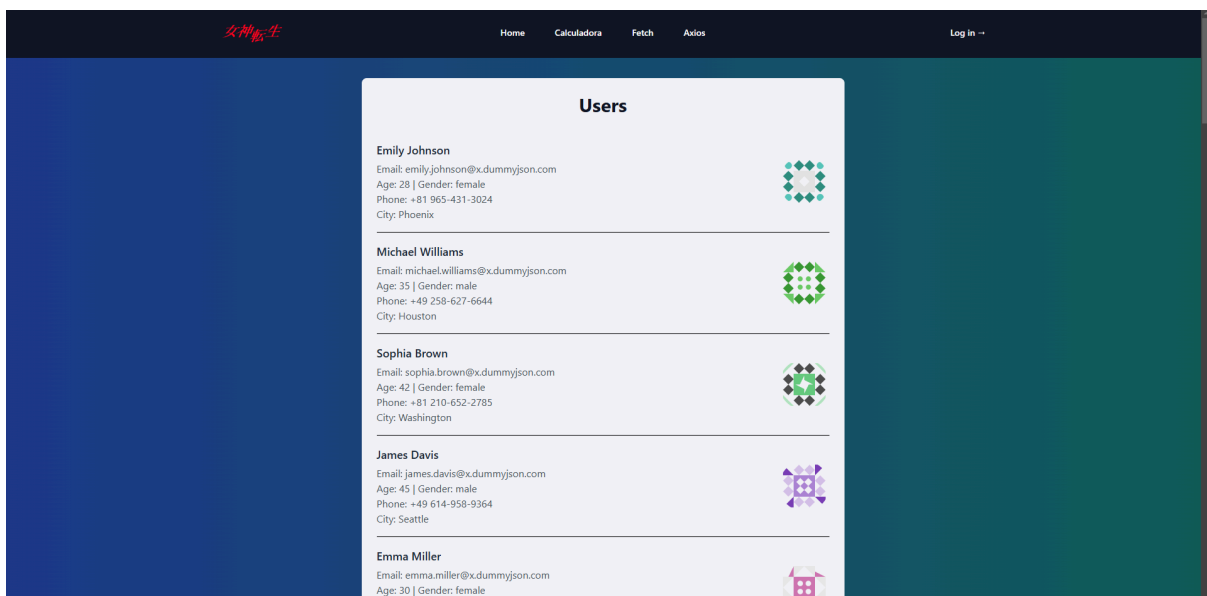
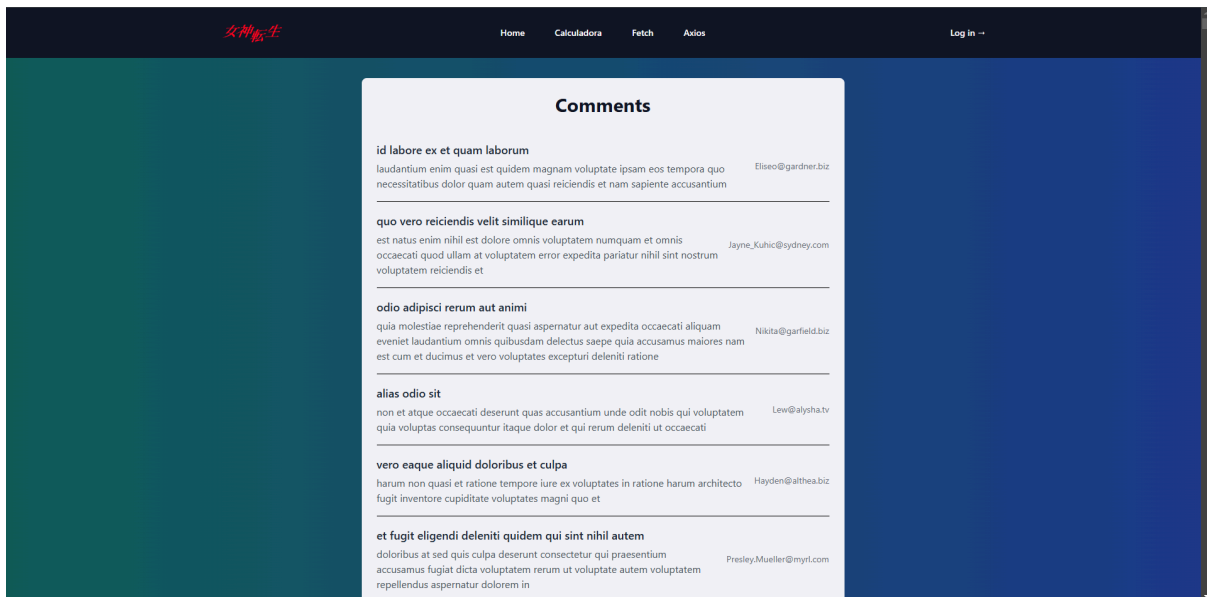
Per a utilitzar la api amb axios faig el mateix però de forma asíncrona i sense convertir a json abans.

El que fa això és es cridar async i quan arriba al await espera a que retorne un valor, així fins que acabe, en compte de fer-ho tot d'una va mostrant poc a poc.

```
1  <script setup> Show component usages
2  import { onMounted, ref } from 'vue'
3  import axios from 'axios'
4
5  const users = ref({value: []})
6
7  onMounted( hook: async () => {
8    try {
9      const response = await axios.get( url: 'https://dummyjson.com/users')
10     users.value = response.data.users
11   } catch (error) {
12     console.log(error)
13   }
14 } )
15 </script>
16
17 <template>
18 <div class="pb-4">
19   <div class="max-w-3xl mx-auto mt-8 p-6 bg-gray-100 shadow-md rounded-lg">
20     <h1 class="text-3xl font-bold text-center text-gray-900 mb-6">Users</h1>
21     <ul class="divide-y divide-black">
22       <li
23         v-for="user in users"
24         :key="user.id"
25         class="py-4 flex flex-col sm:flex-row sm:items-center">
26         <div class="flex-1">
27           <h2 class="text-lg font-semibold text-gray-800">{{ user.firstName }} {{ user.lastName }}</h2>
28           <p class="text-gray-600 mt-1">Email: {{ user.email }}</p>
29           <p class="text-gray-600">Age: {{ user.age }} | Gender: {{ user.gender }}</p>
30           <p class="text-gray-600">Phone: {{ user.phone }}</p>
31           <p class="text-gray-600">City: {{ user.address.city }}</p>
32         </div>
33         
34       </li>
35     </ul>
36   </div>
37 </div>
38 </template>
```

Resultat





TESTS

Per als tests primer instal·lo test-utils, vitest i jsdom.

```
PS D:\Programacion\ProjecteVUE> npm install --save-dev jest @vue/test-utils
```

Install npm module 'vitest'. Click to see all running background tasks

Installing vitest with npm... 3:42 CRLF UTF-8

```
PS D:\Programacion\ProjecteVUE> npx vitest
MISSING DEPENDENCY Cannot find dependency 'jsdom'

✓ Do you want to install jsdom? ... yes

added 29 packages, and audited 479 packages in 4s

73 packages are looking for funding
```

Al package.json afegeixo test: jest.

```
"test": "jest"
```

I al vite.config.js afegeixo que l'entorn a on s'executa sigui jsdom

```
test: {
  environment: 'jsdom',
  include: [
    'src/tests/**/*.test.js'
  ]
}
```

Test de la calculadora:

El que faig és cridar la vista Calculadora i després faig diferents tests que simulen el click al botó.

```

1 import { mount } from '@vue/test-utils';
2 import Calculadora from '@views/Calculadora.vue';
3 import { describe, it, expect } from 'vitest';
4
5 describe('Calculadora', () => { new *
6   it('realitza la suma de 2 + 2', async () :Promise<void> => {
7     const wrapper :VueWrapper<...>, ComponentPublicInstance<...> = mount(Calculadora);
8     await wrapper.find( selector: '[data-testid="digit-2"]' ).trigger( eventString: 'click' );
9     await wrapper.find( selector: '[data-testid="operation-add"]' ).trigger( eventString: 'click' );
10    await wrapper.find( selector: '[data-testid="digit-2"]' ).trigger( eventString: 'click' );
11    await wrapper.find( selector: '[data-testid="equal"]' ).trigger( eventString: 'click' );
12
13    expect(wrapper.text()).toContain( item: '4' );
14  });
15
16   it('realitza la resta de 5 - 3', async () :Promise<void> => {
17     const wrapper :VueWrapper<...>, ComponentPublicInstance<...> = mount(Calculadora);
18     await wrapper.find( selector: '[data-testid="digit-5"]' ).trigger( eventString: 'click' );
19     await wrapper.find( selector: '[data-testid="operation-subtract"]' ).trigger( eventString: 'click' );
20     await wrapper.find( selector: '[data-testid="digit-3"]' ).trigger( eventString: 'click' );
21     await wrapper.find( selector: '[data-testid="equal"]' ).trigger( eventString: 'click' );
22
23     expect(wrapper.text()).toContain( item: '2' );
24   });

```

A calculadora.vue ficaré data-testid per poder simular el click al botó.

```

<button @click="inputDigit( digit: '4' )" class="bg-green-500 text-white text-2xl p-4 rounded-md" data-testid="digit-4">4</button>

```

Per a fer el fetch primer creo un json simulat amb informació real de la api.

```

1 import { mount } from '@vue/test-utils';
2 import { it, expect, beforeEach, vi } from 'vitest';
3 import APIFetch from '@views/APIFetch.vue';
4
5 // Mock de fetch
6 global.fetch = vi.fn( implementation: () :Promise<Awaited<...>> =>
7   Promise.resolve( value: {
8     json: () :Promise<Awaited<...>> => Promise.resolve( value: [
9       {
10         "postId": 1,
11         "id": 1,
12         "name": "id labore ex et quam laborum",
13         "email": "Eliseo@gardner.biz",
14         "body": "laudantium enim quasi est quidem magnam voluptate ipsam eos\ntempora quo necessitatibus\ndolor quam autem quasi\nreic
15       },
16       {
17         "postId": 1,
18         "id": 2,
19         "name": "quo vero reiciendis velit similique earum",
20         "email": "Jayne_Kuhic@sydney.com",
21         "body": "est natus enim nihil est dolore omnis voluptatem numquam\nnet omnis occaecati quod ullam at\ndolore voluptatem error expedita
22       },
23     ]
24   })
25 );

```

Després faig la prova que retorna el contingut de la api.

```
27 // Neteja del mock abans de cada test
28 beforeEach( fn: () :void => { new *
29   fetch.mockClear();
30 });
31
32 it('carrega els comments correctament', async () :Promise<void> => { new *
33   const wrapper : VueWrapper<...>, ComponentPublicInstance<...> = mount(APIFetch);
34   await wrapper.vm.$nextTick();
35
36   await new Promise( executor: resolve => setTimeout(resolve, timeout: 100)); // Espera per simular la càrrega
37
38   const comments : (DOMWrapper<...>[]) = wrapper.findAll( selector: 'li');
39   expect(comments.length).toBe( expected: 2); // Comprova que hi ha dos comentaris
40
41   // Comprova el contingut del primer comentari
42   expect(comments[0].text()).toContain( item: 'id labore ex et quam laborum');
43   expect(comments[0].text()).toContain( item: 'Eliseo@gardner.biz');
44   expect(comments[0].text()).toContain( item: 'laudantium enim quasi est quidem magnam voluptate ipsam eos');
45
46   // Comprova el contingut del segon comentari
47   expect(comments[1].text()).toContain( item: 'quo vero reiciendis velit similique earum');
48   expect(comments[1].text()).toContain( item: 'Jayne_Kuhic@sydney.com');
49   expect(comments[1].text()).toContain( item: 'est natus enim nihil est dolore omnis voluptatем numquam');
50 });
51
```

Per a axios faig el mateix però modificant en contingut de la api simulada.

```
1 import { mount } from '@vue/test-utils'; // permet muntar els components fer proves
2 import { describe, it, expect, vi } from 'vitest';
3 import axios from 'axios';
4 import APIAxios from '@views/APIAxios.vue';
5
6 vi.mock( path: 'axios');
7
8 describe('APIAxios.vue', () :void => { new *
9   it('carrega els users correctament', async () :Promise<void> => {
10     const mockUsers : {users: [{firstName: string, lastName:...}] = {
11       users: [
12         {
13           id: 1,
14           firstName: "Emily",
15           lastName: "Johnson",
16           age: 28,
17           gender: "female",
18           email: "emily.johnson@x.dummyjson.com",
19           phone: "+81 965-431-3024",
20           address: {
21             city: "Phoenix"
22           },
23           image: "https://dummyjson.com/icon/emilys/128"
24         },
25         {
26           id: 2,
27           firstName: "Michael",
28           lastName: "Williams",
29           age: 35,
30           gender: "male",
31           email: "michael.williams@x.dummyjson.com",
32           phone: "+49 258-627-6644",
```

```
41   axios.get.mockResolvedValueOnce({ data: mockUsers });
42   const wrapper : VueWrapper<...>, ComponentPublicInstance<...> = mount(APIAxios);
43   await wrapper.vm.$nextTick();
44
45   await new Promise( (executor, resolve => setTimeout(resolve, timeout: 100));
46
47   const users : (DOMWrapper<...>>[] = wrapper.findAll( selector: 'li');
48   expect(users.length).toBe( expected: 2);
49   expect(users[0].text()).toContain( item: 'Emily Johnson');
50   expect(users[0].text()).toContain( item: 'Email: emily.johnson@x.dummyjson.com');
51   expect(users[0].text()).toContain( item: 'City: Phoenix');
52
53   expect(users[1].text()).toContain( item: 'Michael Williams');
54   expect(users[1].text()).toContain( item: 'Email: michael.williams@x.dummyjson.com');
55   expect(users[1].text()).toContain( item: 'City: Houston');
56
57   const userImages : (DOMWrapper<...>>[] = wrapper.findAll( selector: 'img');
58   expect(userImages.length).toBe( expected: 2);
59   expect(userImages[0].attributes( key: 'src')).toBe( expected: 'https://dummyjson.com/icon/emilys/128');
60   expect(userImages[1].attributes( key: 'src')).toBe( expected: 'https://dummyjson.com/icon/michaelw/128');
61   });
62   });
```

Després executo els tests utilitzant npx vitest.

```
RERUN rerun all tests
```

```
✓ src/tests/Axios.test.js (1)
✓ src/tests/Calculadora.test.js (13)
✓ src/tests/Fetch.test.js (1)
```

```
Test Files  3 passed (3)
Tests       15 passed (15)
Start at    19:51:54
Duration    861ms
```

```
PASS Waiting for file changes...
press h to show help, press q to quit
```



Render

Projecte estàtic muntat al render:

