



Grupo 3

Martín Ocqueteau

Nicolás Fraga

Agustín Ríos

1

Consulta a APIs de Chistes 

3 Tipos

- Dad Jokes 
- Geek Jokes 
- F-Off 

Demostración



I don't trust stairs. They're always up to something.



Who the fuck are you anyway, Mr. Lebowski,
why are you stirring up so much trouble, and,
who pays you?

- Other Lebowski



No statement can catch the
`ChuckNorrisException`.

2

Frontend ✨

Options API vs Composition API

```
<script lang="ts">
import { defineComponent } from "vue";
export default defineComponent({
  data() {
    return {
      hello: "hello",
    }
  }
})
</script>
<template>
  <h1>{{ hello }}</h1>
</template>
<style scoped>
h1 {
  color: red;
}
</style>
```

hello

```
<script setup lang="ts">
import { ref } from 'vue';
const hello = ref('hello');
</script>
<template>
  <h1>{{ hello }}</h1>
</template>
<style scoped>
h1 {
  color: red;
}
</style>
```

AddFavouriteButton

```
<script setup lang="ts">
defineProps({
  handleClick: Function,
});
</script>

<template>
  <button @click="handleClick ? handleClick() : null">
    Añadir a favoritos
  </button>
</template>
```

dom Dad Joke

oke

Añadir a favoritos

JokeCard

Encargado de mostrar el texto pedido a las APIs en una variable.

Variable actualizada con los métodos de botones. Se actualiza el Prop en base a la promesa de consultar a la API.

JokeCard

```
<script setup  
  lang="ts">  
  defineProps({  
    joke: String,  
  });  
</script>
```

```
<template>  
  <div class="joke-card">  
    <p>  
      <strong>{{ joke || "Request a joke!" }}</strong>  
    </p>  
  </div>  
</template>
```

FOAASCard

Otro approach para lograr manejo reactivo de strings.

Se utilizan los campos (fields) que la api requiere para entregar un F-Off.

Se utilizan los campos (fields) que la api requiere para entregar un F-Off.

FOAASCard

```
<script>
defineProps({
  fields: Array,
  foTextArray: Array,
});
</script>
```

```
<template>
  <div class="joke-card">
    <p>
      <strong>
        {{ concatIntoStr(foTextArray, fields) || "Request a
joke!" }}
      </strong>
    </p>
  </div>
</template>
```

FOAASCard

concatIntoStr(foTextArray, fields)

???

1. **foTextArray**: Respuesta en texto de la API
2. **fields**: información de la operación a la que llamamos, proporcionada por API.

1.

```
foTextArray = ":name, WTF?"
```

2.

```
fields = [  
  {name: 'Name',  
   field: 'name'},  
  
  {name: 'From',  
   field: 'from'}  
]
```

FOAASCard

concatIntoStr(foTextArray, fields)

???

1. **foTextArray**: Respuesta en texto de la API
2. **fields**: información de la operación a la que llamamos, proporcionada por API.

1.

```
foTextArray = ":name, WTF?"
```

2.

```
fields = [  
  {name: 'Name',  
    field: 'name',  
    text: ':name'},  
  
  {name: 'From',  
    field: 'from',  
    text: ':from'}  
]
```

FOAASCard

Uso de v-model en vista que contiene a componente.

```
<div v-for="field in fields">
  <div class="field-line">{{field.name}}:
  <input class="field-text-box" v-model="field.text">
  </div>
</div>
```

F-Off Generator

Load F-Off

Añadir a favoritos

Name:

From:

FOAASCard

```
{name: 'Name', field: 'name', text: ':name'}
```

:name you're a fucking Rock Star!

- :from

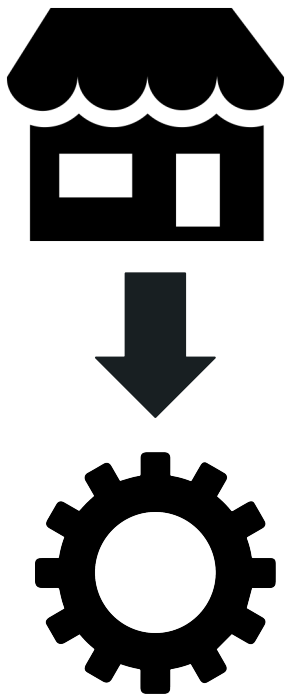
```
{name: 'From', field: 'from', text: ':from'}
```

3

Manejo de estado



Store



```
import { defineStore } from "pinia";

// Declaramos los tipos de elementos que guarda el estado
interface stateInterface {
  jokes: string[];
}

export const useJokeStore = defineStore({
  id: "favourite-jokes",
  state: () => ({ jokes: [] } as stateInterface),
  getters: {
    length(): number {
      return this.jokes.length;
    },
  },
  actions: {
    save(jokeText: string) {
      this.jokes.push(jokeText);
    },
    delete(joke: string) {
      this.jokes = [...this.jokes.filter((j) => j !== joke)];
    },
  },
});
```

```
type str_enumerator = Generator<[number, string]>;

const jokeStore = useJokeStore();
const jokesLength = jokeStore.length;
const sIfPlural = ref(jokesLength === 1 ? "joke" : "jokes");
const indexedJokes = enumerate(jokeStore.$state.jokes) as str_enumerator;
```

4

Lo que aprendimos 🤔