



IIIC3548 - DAAW

## Tarea 6 - Svelte

Grupo 7: Matías Cadile, Matías Soto y José Luco

# Nuestra app

CATS V/S DOGS

Quedan **4** segundos!



Votar

**0 VOTES**



Votar

**0 VOTES**

IT'S A TIE

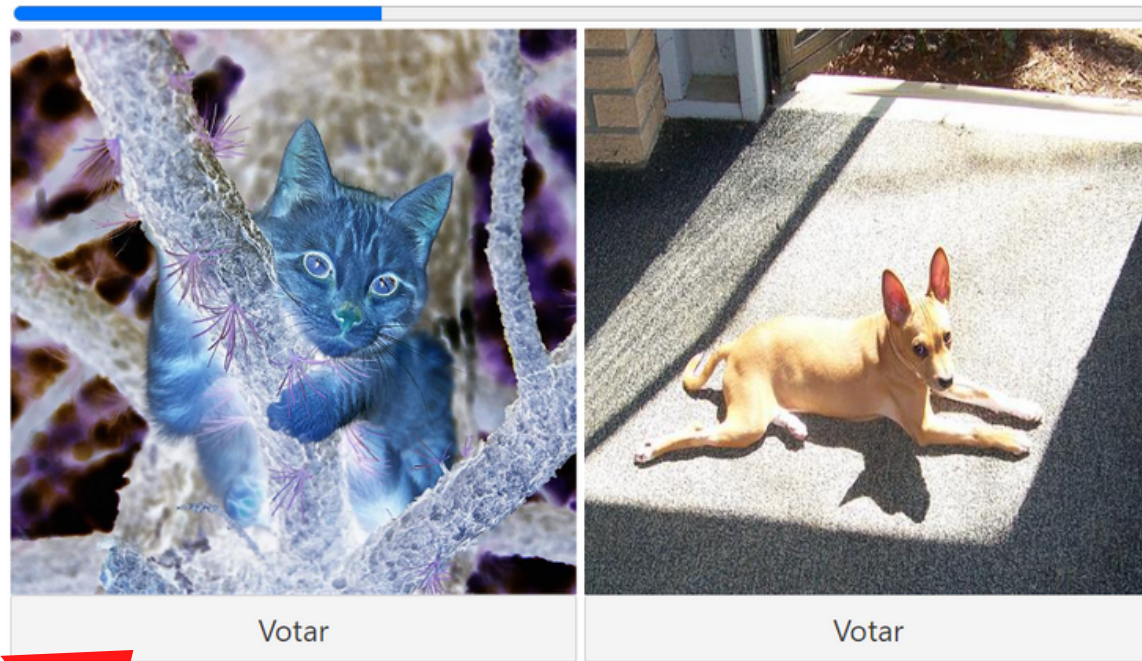
DEMO

# Estructura y eventos principales

## Counter

CATS V/S DOGS

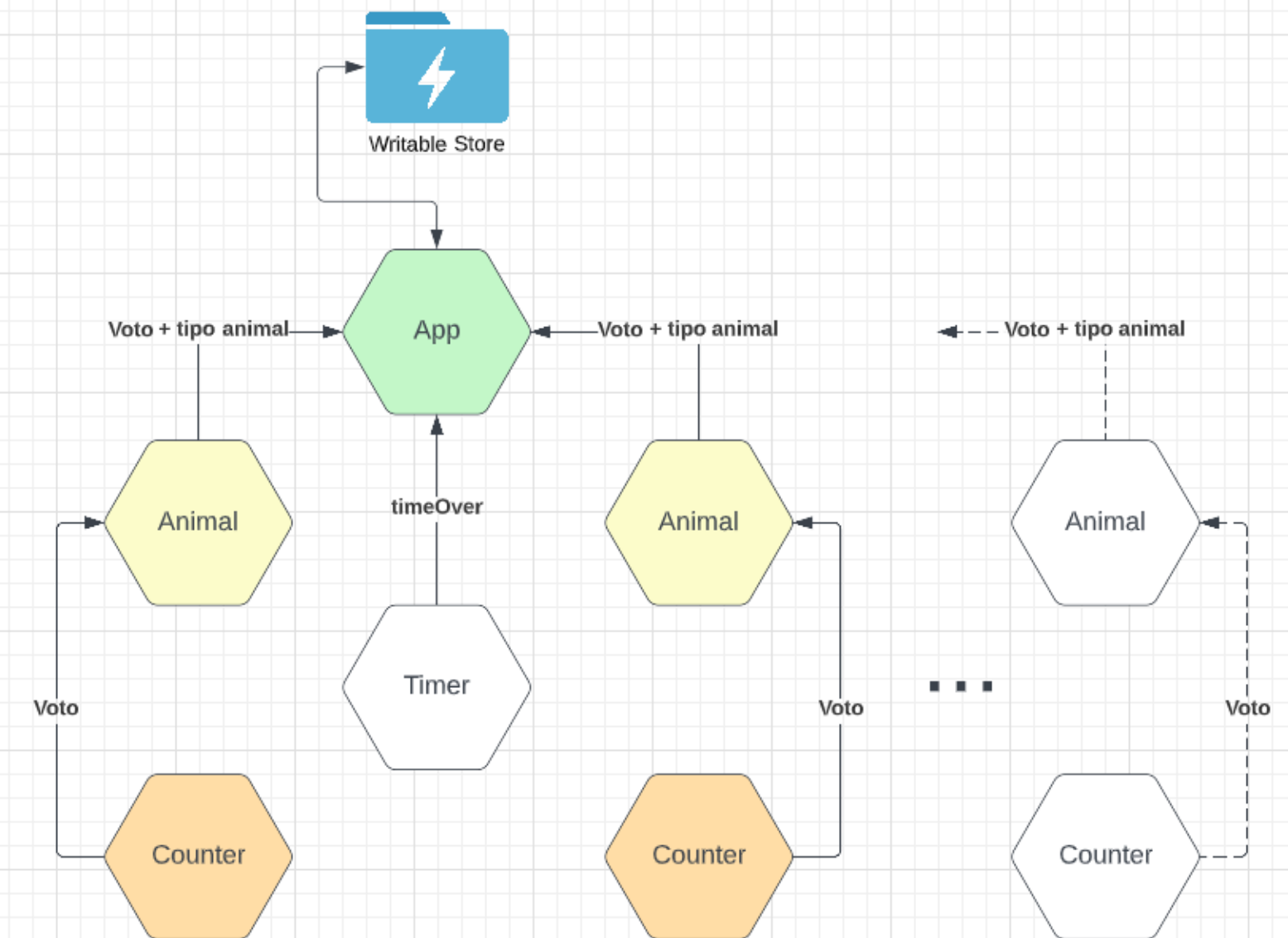
Quedan 6 segundos!



0 VOTES

0 VOTES

IT'S A TIE



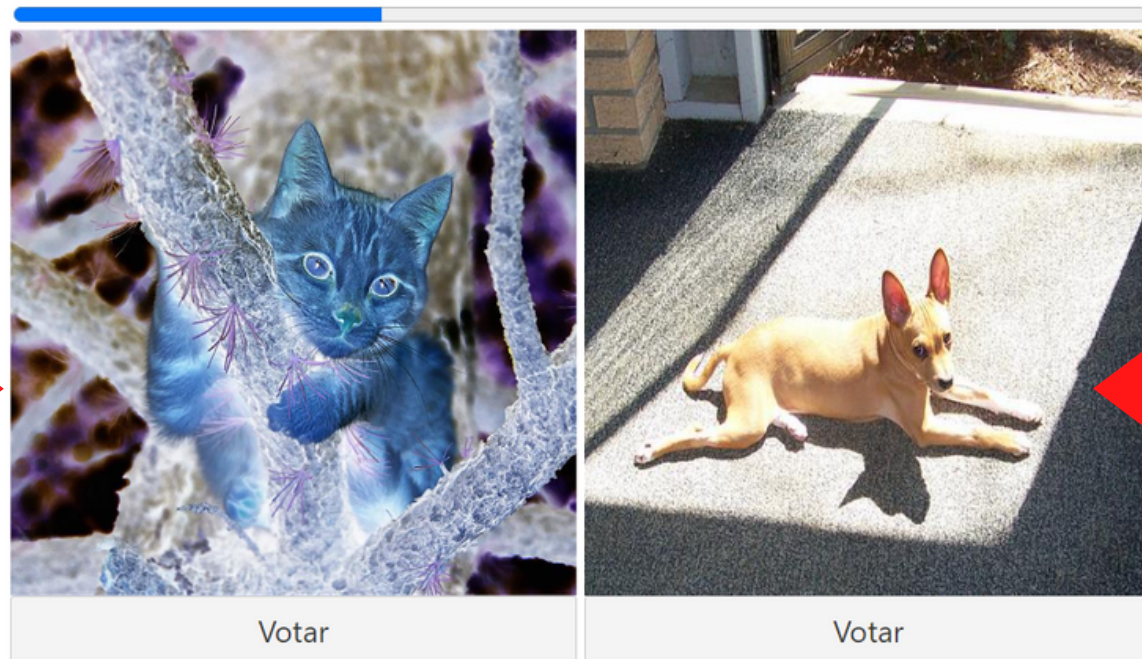


# Estructura y eventos principales

## Animal

CATS V/S DOGS

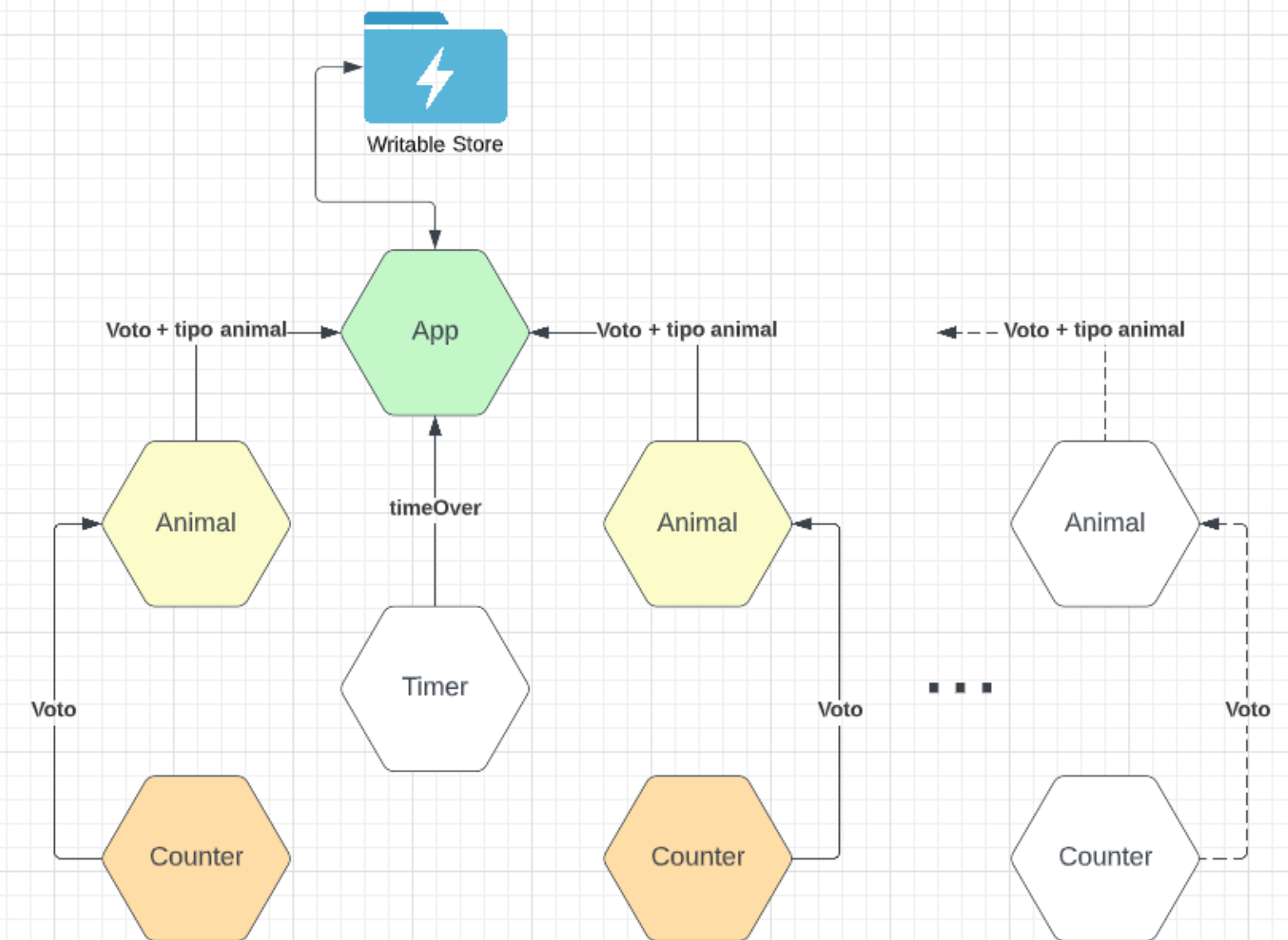
Quedan 6 segundos!



0 VOTES

0 VOTES

IT'S A TIE

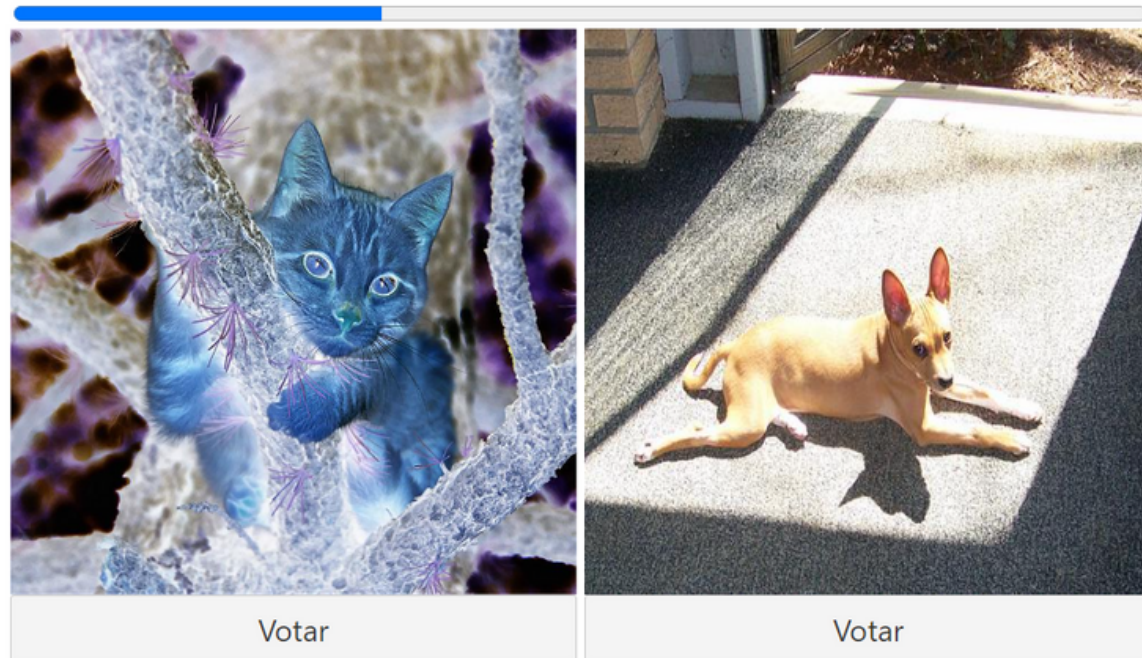


# Estructura y eventos principales

## Timer

CATS V/S DOGS

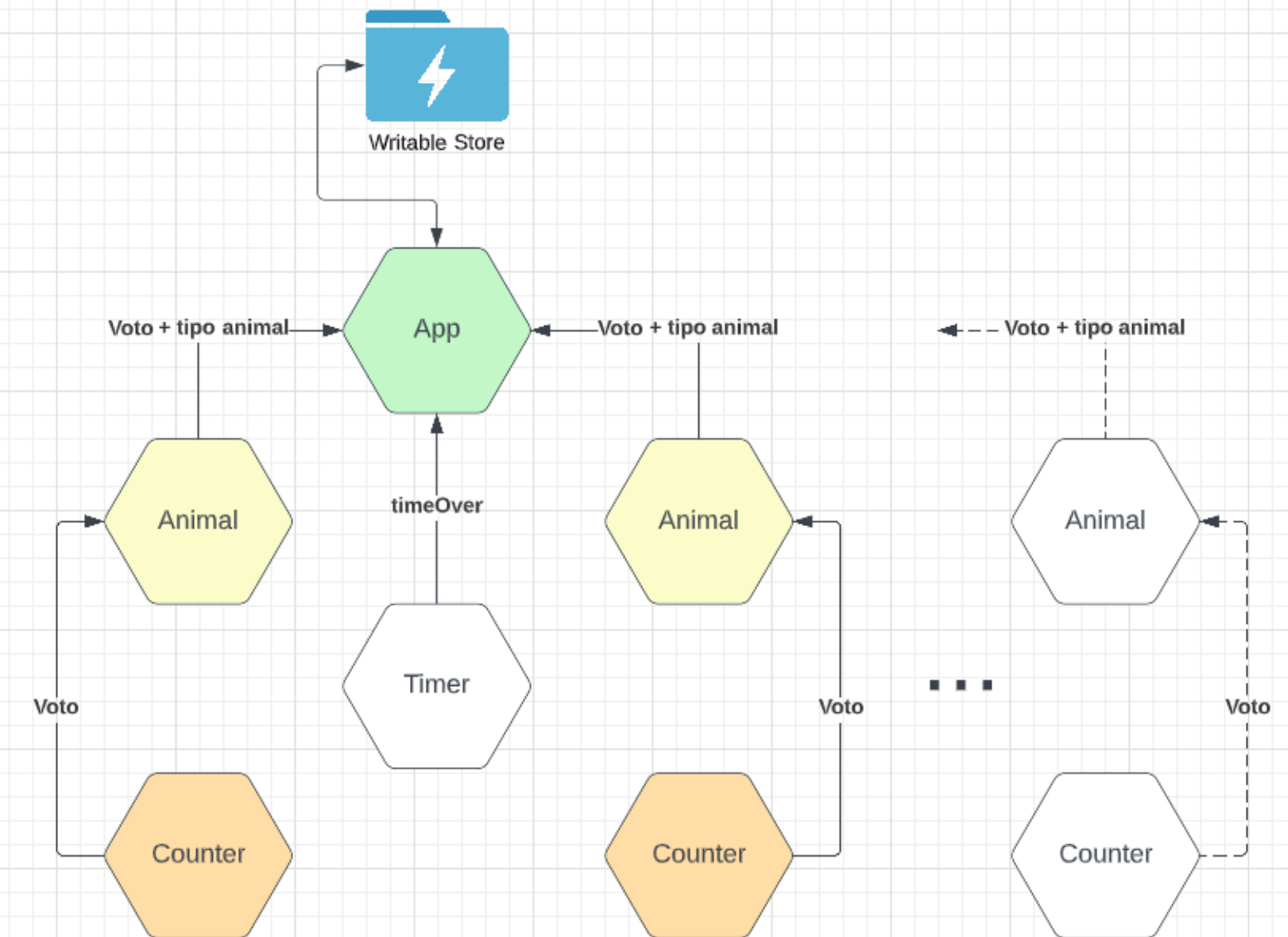
Quedan 6 segundos!



0 VOTES

0 VOTES

IT'S A TIE



# Custom Event Handling / Forwarding

- A diferencia de los eventos en el DOM, los eventos entre Componentes no "suben" como burbujas



Si queremos escuchar eventos desde componentes anidados: Forwarding

# Custom Event Handling / Forwarding

Counter.svelte



```
const addVote = ()=>{  
  dispatch('vote', {  
    value: 1  
  });  
}  
  
<button on:click={addVote}>  
  Votar  
</button>
```



# Custom Event Handling / Forwarding

## Counter.svelte

```
const addVote = () => {
  dispatch('vote', {
    value: 1
  });
}

<button on:click={addVote}>
  Votar
</button>
```



## Animal.svelte

```
<div class="animal">
  {#if src}
    <img {src} alt='random animal'>
  {:else}
    <p>Loading...</p>
  {/if}

  <Counter on:vote={forward}></Counter>
  <p><strong> {currentCount} Votes</strong></p>
</div>
```



```
const forward = () => {
  getAnimalPicture();
  dispatch("vote", {value:event.detail, animal: data['animal']});
};
```

# Custom Event Handling / Forwarding

Animal.svelte

```
<div class="animal">
  {#if src}
    <img {src} alt='random animal'>
  {:else}
    <p>Loading...</p>
  {/if}
  <Counter on:vote={forward}></Counter>
  <p><strong> {currentCount} Votes</strong></p>
</div>
```

```
const forward = () => {
  getAnimalPicture();
  dispatch("vote", {value:event.detail, animal: data['animal']});
};
```

```
const voteReceived = (vote) => {
  console.log("vote received for: ", vote.detail.animal )
  if (vote.detail.animal == 'cat') {
    cat.update((currentCatVotes) => {
      return currentCatVotes + vote.detail.value
    })
  } else {
    dog.update((currentDogVotes) => {
      return currentDogVotes + vote.detail.value
    })
  }
}

<div class="votes">
  {#if toggleCat}
    <Animal on:vote={voteReceived} data={catData}></Animal>
  {/if}
  {#if toggleDog}
    <Animal on:vote={voteReceived} data={dogData}></Animal>
  {/if}
</div>
```

App.svelte

# Manejo de estados

# Manejo de estados

## Writable Stores

store.js

```
import { writable } from "svelte/store";

export const cat = writable(0);
export const dog = writable(0);
```

App.svelte

```
const catUnsubscribe = cat.subscribe(value => {
  catVotes = value;
});
const dogUnsubscribe = dog.subscribe(value => {
  dogVotes = value;
});

const voteReceived = (vote) => {
  console.log("vote received for: ", vote.detail.animal);
  if (vote.detail.animal === 'cat') {
    cat.update((currentCatVotes) => {
      return currentCatVotes + vote.detail.value;
    });
  } else {
    dog.update((currentDogVotes) => {
      return currentDogVotes + vote.detail.value;
    });
  }
}
```

```
{#if catVotes > dogVotes }
  {#if confettiVisible}
    // CATS WIN!
  {/if}
{:else if dogVotes > catVotes}
```

```
.
.
.
.
```

# Dificultades

- Poca documentación / soluciones en foros
- *Workarounds* en funcionalidades "base"
- *Readable store para Timer: pendiente*

# Conclusiones

- Baja curva de aprendizaje se contradice con la información / ejemplos disponibles
- Rápido y liviano



**Muchas gracias**