

# PWA

## Progresie Web Apps

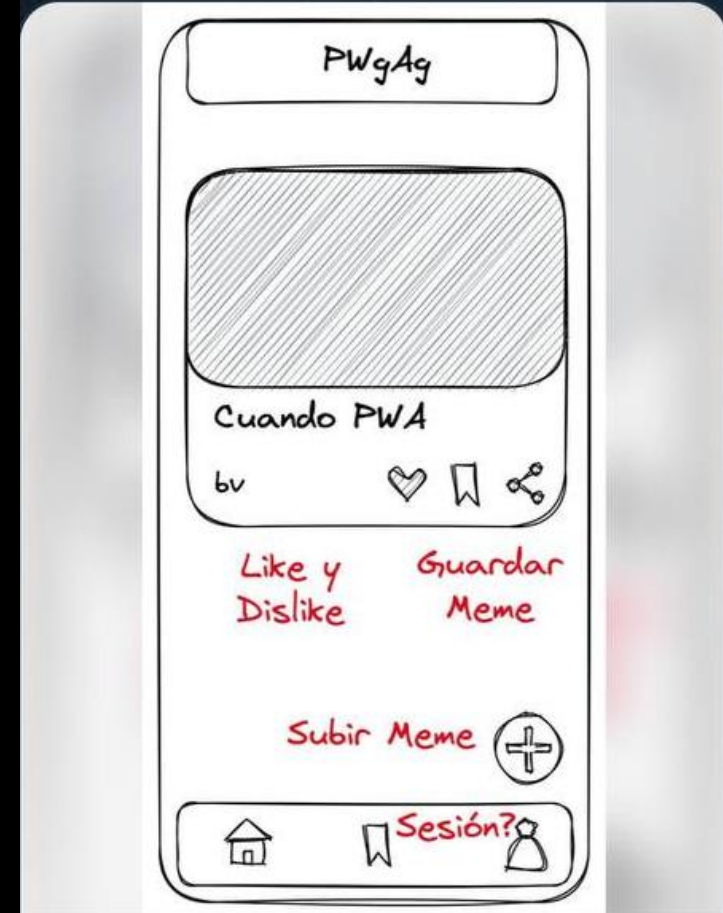
Benjamín  
Vicente

Jose Antonio  
Castro

José  
Madriaza

# Planificación

- Un mensaje de Telegram



- usuarios
- datos de memes (idealmente guardando fotos)
- datos de likes
- push cuando alguien te hace like a un meme
- guardar meme
- offline funciona el layout y los memes guardados

# Demo

PWgAg



Baelfire18

hace 4 días

**nyan cat**



1 like



AlfreMK

hace 4 días

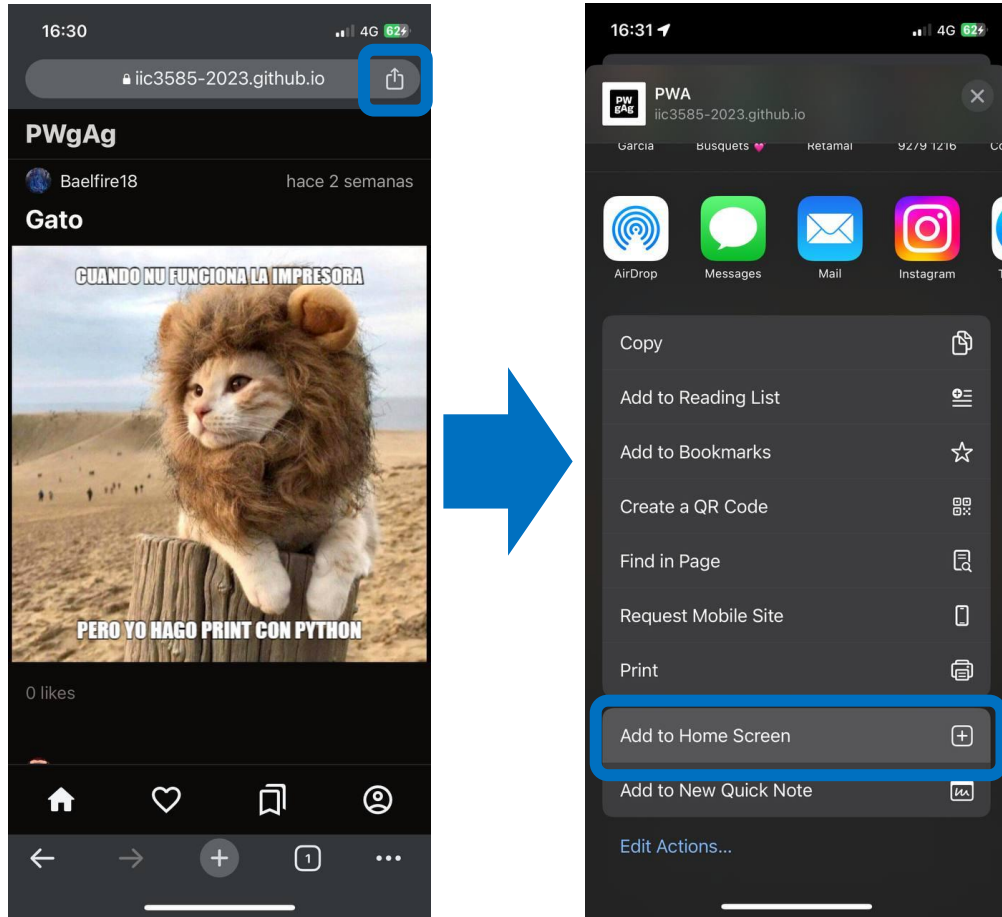
**perro meme**



# Firebase

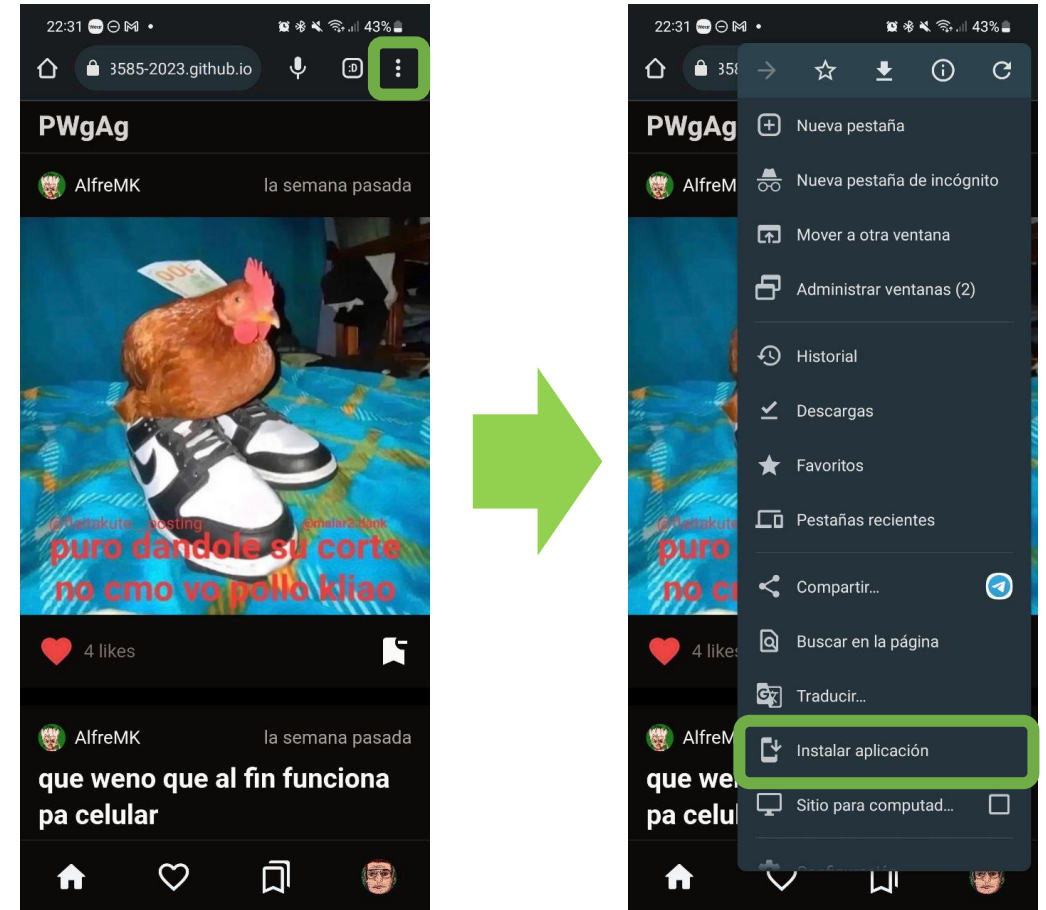
- Autenticación 
- Realtime Database 
- Almacenamiento de imágenes 
- Notificaciones push con Messaging y Cloud Functions  

# Instalable



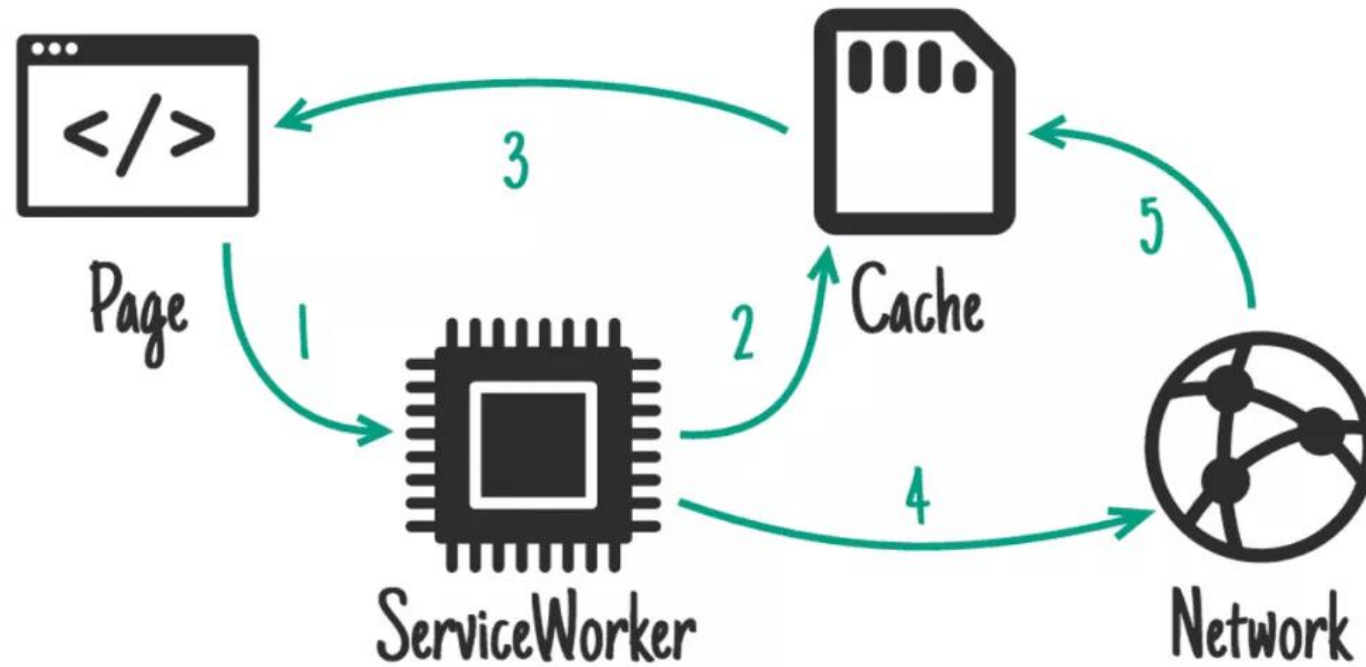
Instalación en iOS

VS




Instalación en Android

# Service Worker Cache



# IndexedDB storage

PWgAg



Storage

- Local Storage
- Session Storage
- IndexedDB
  - firebase-heartt
  - firebase-install
  - firebase-mess
  - firebaseLocalS
  - main - https://i
    - images
    - posts
  - firebase-heartt
  - firebase-install

Total entries: 1

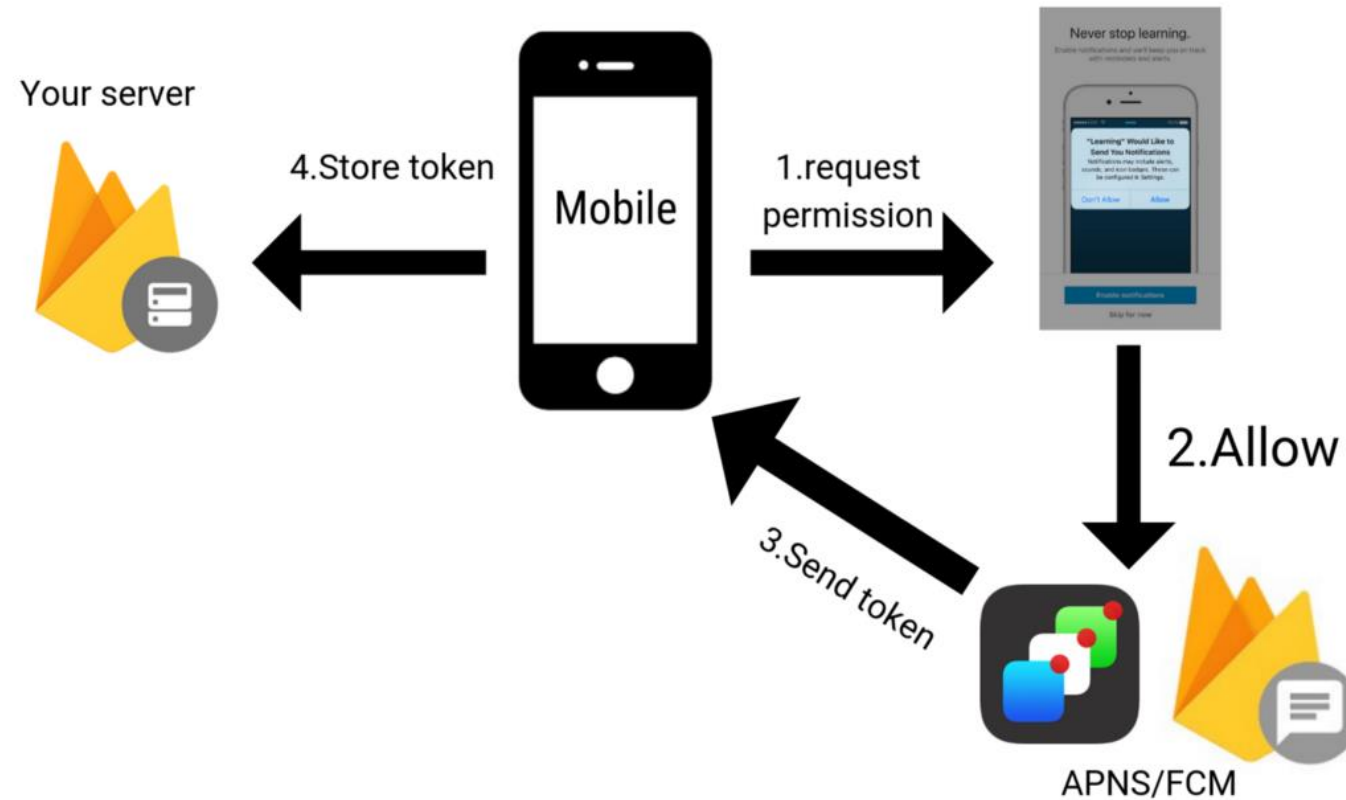
#	Key (Key path: "key")	Value
0	"1684724861550"	{key: '1684724861550', authorUsern authorUserName: "benjavicente" description: "Balu" key: "1684724861550" resourceURL: "https://firebasest

```
▼ blob: Blob  
  size: 100565  
  type: "image/jpeg"  
  url: "https://firebasest  
      image/jpeg
```



```
1 const request = indexedDB.open(name, version);  
2 request.onupgradeneeded = (event) => onUpgradeNeeded(event, request.result);  
3 request.onsuccess = () => resolve(request.result);  
4 request.onerror = () => reject(request.error);
```

# Notifications API



\*Corríjanme si el flujo no funciona así



# Notifications API

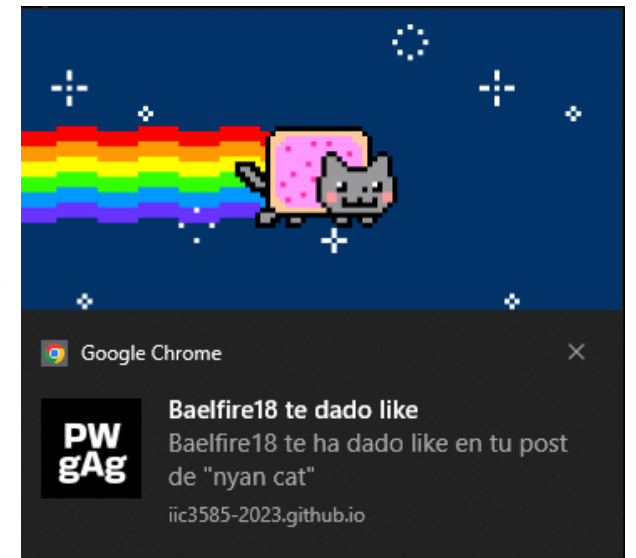
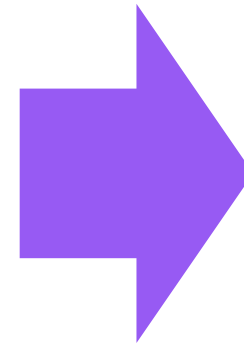


```
const app = admin.initializeApp();
const db = admin.database(app);
const msg = admin.messaging(app);

export const onLike = onValueWritten("posts/{postID}/likes/{userID}", async (event) => {
  // Se obtiene el dueño del post
  const { authorUserName, description, resourceURL }: Post = postSnap.val();

  // Se obtiene la lista de tokens del dueño
  const tokens: string[] = Object.keys(tokensSnap.val());

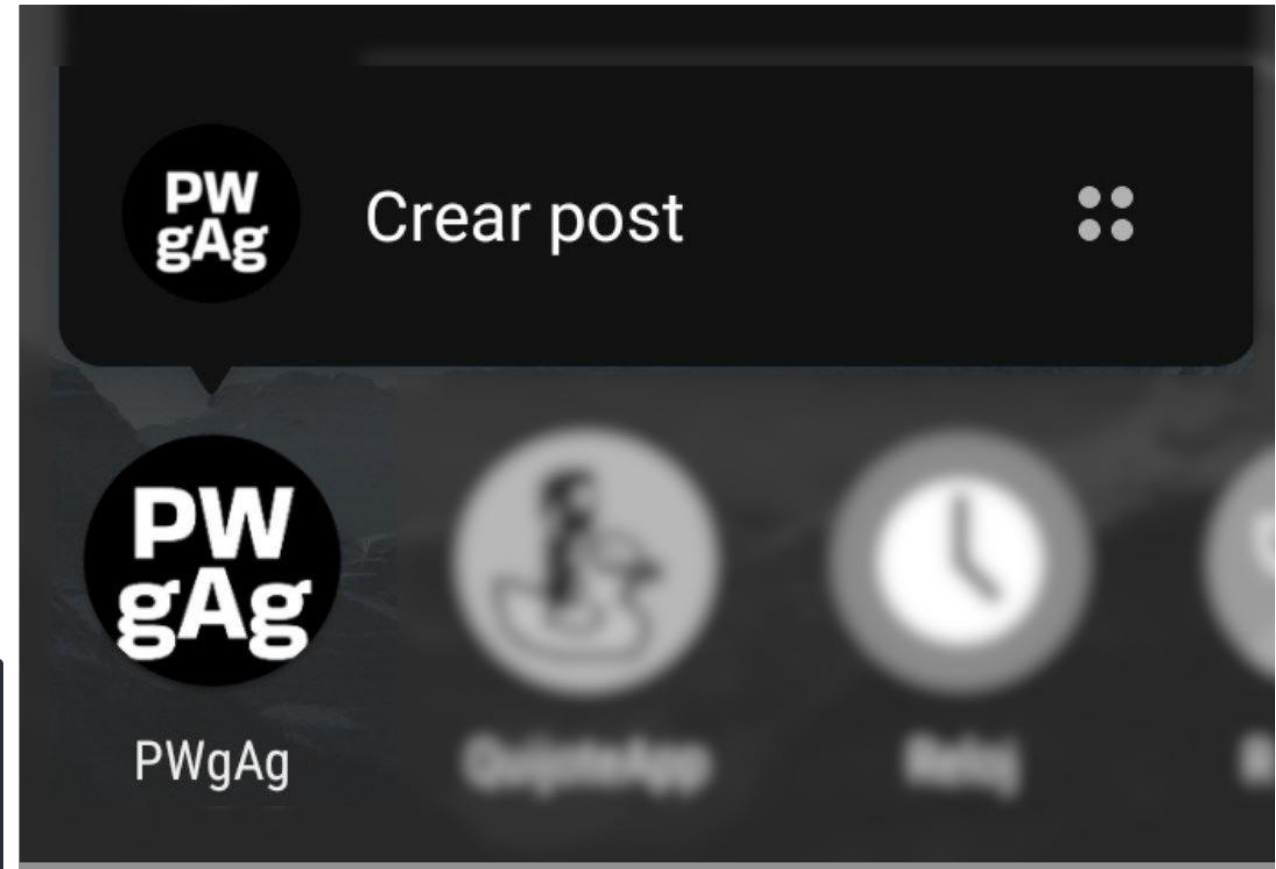
  // Se envía la notificación
  const title = `${userID} te dado like`;
  const body = `${userID} te ha dado like en tu post de "${description}"`;
  await msg.sendEachForMulticast({
    tokens,
    notification: { title, body, imageUrl: resourceURL },
    webpush: {
      notification: { title, body, icon: resourceURL },
    },
  });
});
```



# App shortcuts

Abrir la app en una url en específico

```
"shortcuts": [  
  {  
    "name": "Crear post",  
    "short_name": "Crear post",  
    "url": "/pwa-grupo-01/?create",  
    "description": "Crear post",  
    "icons": [  
      {  
        "src": "/pwa-grupo-01/icons/PWgAg-192.png",  
        "type": "image/png",  
        "sizes": "192x192"  
      }  
    ]  
  }  
]
```



```
if (window.location.search === "?create") {  
  setTimeout(() => {  
    console.log("open create modal");  
    setPageIndex(0);  
    getElById("create-btn").dispatchEvent(new Event("click"));  
  }, 0);  
}
```

# View Transitions

Experimental desde 2023

- Animaciones usando el DOM
- Cambios en JS y CSS mínimos

```
document.startViewTransition(() => updateTheDOMSomehow(data))
```

```
#content.to-right {  
  view-transition-name: content-to-right;  
}  
  
::view-transition-new(content-to-right) {  
  animation: 300ms cubic-bezier(0.4, 0, 0.2, 1) both slide-in-from-left;  
}  
  
::view-transition-old(content-to-right) {  
  animation: 300ms cubic-bezier(0.4, 0, 0.2, 1) both slide-out-to-right;  
}
```

Fuente: [https://developer.mozilla.org/en-US/docs/Web/API/View Transitions API](https://developer.mozilla.org/en-US/docs/Web/API/View_Transitions_API)



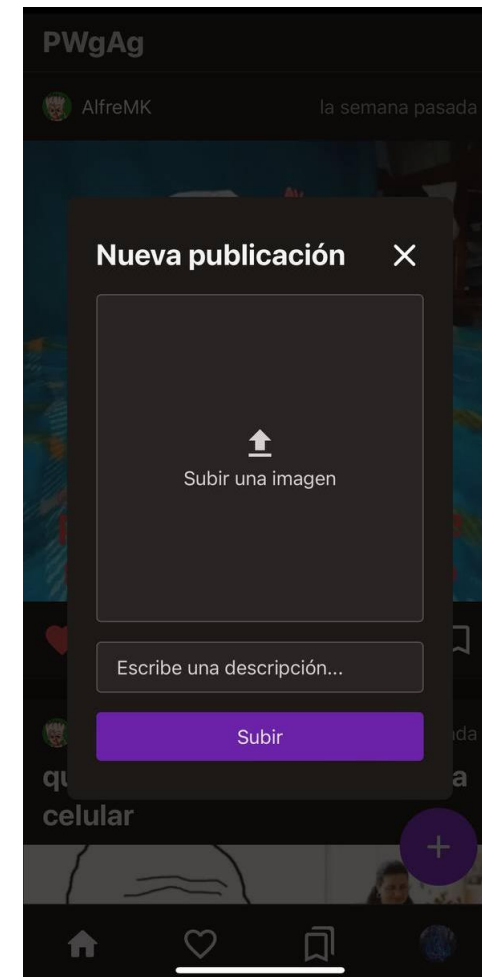
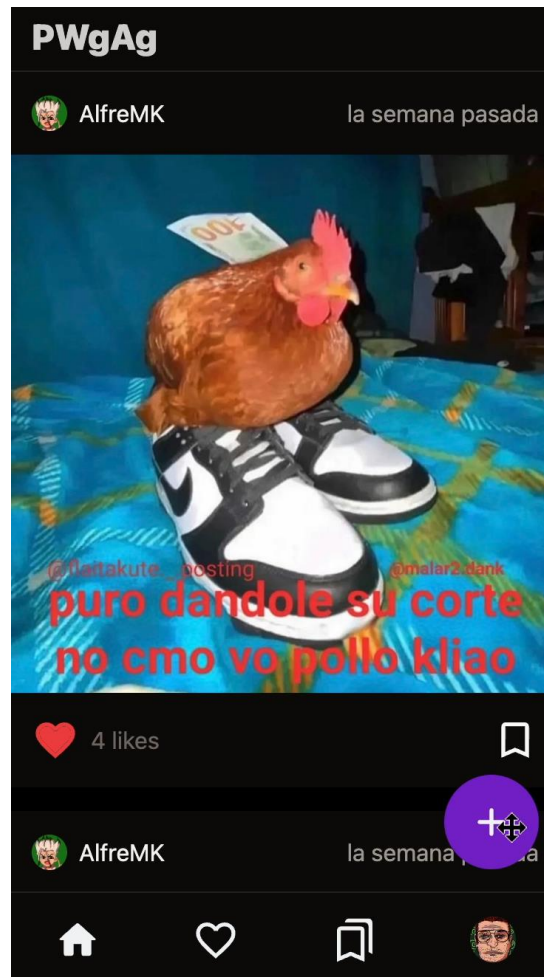
# Dialog Element

Estable desde el 2022

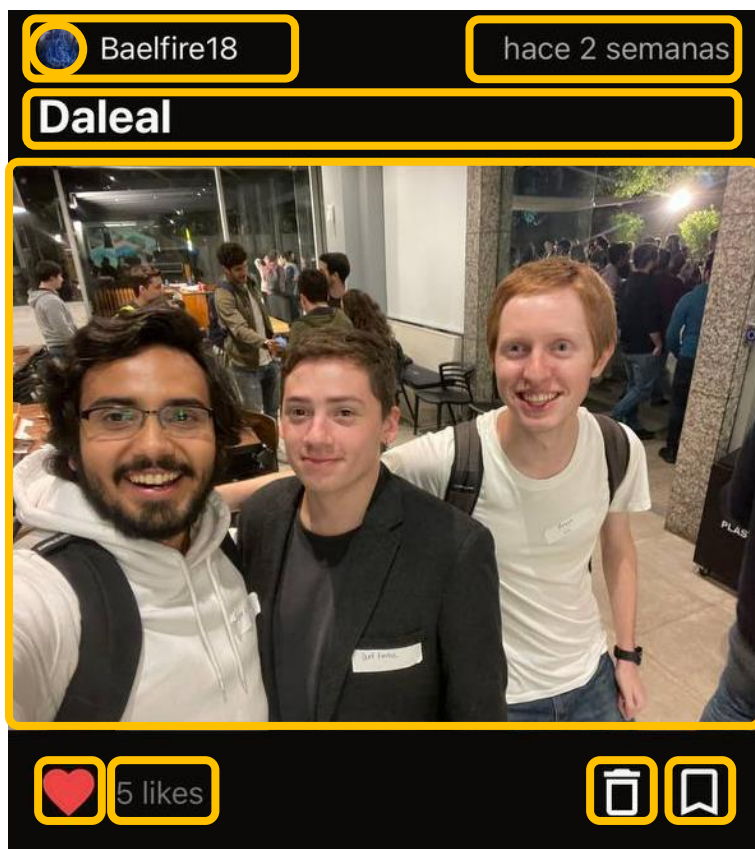
- Backdrop y centrado en capa aparte
- Accesible por defecto

```
<dialog>  
...  
</dialog>
```

```
createDialog.showModal()  
createDialog.close()
```



# Template Element



Fragmentos de HTML reutilizables

```
<li role="article" class="flex w-full flex-col bg-stone-950 py-4">
  <header class="px-4">
    <div class="flex items-center justify-between">
      <div class="flex items-center gap-2">
        <img class="post-owner-avatar h-6 w-6 rounded-full bg-stone-600" />
        <div class="post-owner-name text-stone-100"></div>
      </div>
      <div>
        <time class="post-created-at text-stone-400"></time>
      </div>
    </div>
    <h2 class="post-caption py-2 text-2xl font-bold"></h2>
  </header>
  <figure class="post-figure flex max-h-96 min-h-[18rem] w-full items-center justify-center bg-black">
    <img class="post-img w-full aspect-auto h-full" />
  </figure>
  <footer class="flex items-start justify-between p-4 pb-0">
    <div class="flex items-center gap-2">
      <button class="post-like hide-on-anon">
        <svg xmlns="http://www.w3.org/2000/svg" width="32" height="32" viewBox="0 0 24 24">
          <path
            fill="currentColor"
            d="M16.5 5c-1.54 0-3.04.99-3.56 2.36h-1.87C10.54 5.99 9.04 5 7.5 5C5.5 5 4 6.5 4 8.5c0 2.89 3.14 5.74 7.9 10.05L1.11.11C16.86 14.24 20 11.39 20 8.5c0-2-1.5-3.5-3.5z"
          />
        </svg>
      </button>
      <div class="post-like-count text-stone-500"></div>
    </div>
    <div class="hide-on-anon flex gap-2">
      <button class="post-delete">
        <svg xmlns="http://www.w3.org/2000/svg" width="32" height="32" viewBox="0 0 24 24">
          <path fill="currentColor" d="M6 19c0 1.19 2 2 2 2h8c1.1 0 2-.9 2-2V7h6v12z" />
        </svg>
        <span class="sr-only">Eliminar post</span>
      </button>
      <button class="post-save">
        <svg class="post-add-to-cache-svg" xmlns="http://www.w3.org/2000/svg" width="32" height="32" viewBox="0 0 24 24">
          <path fill="currentColor" d="M17 3H7c-1.1 0-1.99.9-1.99 2L5 21L7 31L7 3V5c0-1.1-.9-2-2-2z" />
        </svg>
        <svg class="post-rm-from-cache" xmlns="http://www.w3.org/2000/svg" width="32" height="32" viewBox="0 0 24 24">
          <path fill="currentColor" d="M21 7h-6V5h6v2z" />
        </svg>
        <span class="post-save-msg sr-only">Guardar</span>
      </button>
    </div>
  </footer>
</li>
```

# Intl.RelativeTimeFormat

Parte de la API de internacionalización  
Estable desde 2020

```
export function getTimeAgo(date) {  
  const seconds = (new Date(+date).getTime() - Date.now()) / 1000;  
  for (const [unit, range] of Object.entries(timeDeltaRanges)) {  
    if (Math.abs(seconds) > range || unit === "seconds") {  
      const value = Math.round(seconds / range);  
      return timeDeltaFormatter.format(value, unit);  
    }  
  }  
  return "ahora";  
}
```

```
const timeDeltaFormatter = new Intl.RelativeTimeFormat("es-cl", { numeric: "auto" });
```



# ESModules

Estable desde 2018

- Permite modularizar mucho más fácilmente el código
- URL imports permiten usar librerías sin bundler

```
import { getMessaging, getToken, onMessage } from "https://www.gstatic.com/firebasejs/9.21.0/firebase-messaging.js";
import { ref as refDB, set as setDB } from "https://www.gstatic.com/firebasejs/9.21.0/firebase-database.js";
import { createSignal, createEffect } from "../ui.js";
import { app, firebaseConfig } from "./app.js";
import { userSignal } from "./auth.js";
import { db } from "./db.js";
import { setPageIndex } from ".././main.js";
import { registrationSignal } from "../registration.js";
import { addNotificationToCache } from ".././localDB.js";
```

# Principales problemas

- Funcionamiento offline
- Cache
- Interceptar notificaciones
- Unir todo 😊