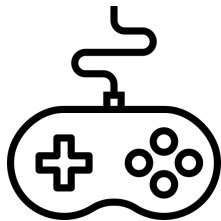




# PWAs

Grupo 6  
Sección 1

Juan Pablo Contreras  
Rodrigo Heiremans  
Matias Perez



# Demo

(solo funciona con chrome)

<https://front-ctplmdc.herokuapp.com/>



# Código



# Manifest.js

## ● Web app manifest and service worker meet the installability requirements

Service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. With proper service worker and manifest implementations, browsers can proactively prompt users to add your app to their homescreen, which can lead to higher engagement. [Learn more.](#)

## ● Configured for a custom splash screen

A themed splash screen ensures a high-quality experience when users launch your app from their homescreens. [Learn more.](#)

## ● Sets a theme color for the address bar.

The browser address bar can be themed to match your site. [Learn more.](#)

```
{
  "name": "Cuando te pasan la musica del carrete",
  "short_name": "CTPLMDC",
  "start_url": ".",
  "background_color": "#6dcdb1",
  "theme_color": "#009578",
  "display": "standalone",
  "icons": [
    {
      "src": "images/manifest-icon-192.maskable.png",
      "sizes": "192x192",
      "type": "image/png",
      "purpose": "any"
    },
    {
      "src": "images/manifest-icon-192.maskable.png",
      "sizes": "192x192",
      "type": "image/png",
      "purpose": "maskable"
    },
    {
      "src": "images/manifest-icon-512.maskable.png",
      "sizes": "512x512",
      "type": "image/png",
      "purpose": "any"
    },
    {
      "src": "images/manifest-icon-512.maskable.png",
      "sizes": "512x512",
      "type": "image/png",
      "purpose": "maskable"
    }
  ]
}
```



# Service Worker

```
if ("serviceWorker" in navigator) {  
  navigator.serviceWorker.register("firebase-messaging-sw.js").then(registration => {  
    console.log(registration);  
  }).catch(error => {  
    console.error(error);  
  });  
}
```



You, 6 hours ago • debugg ...

## ● Web app manifest and service worker meet the installability requirements

Service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. With proper service worker and manifest implementations, browsers can proactively prompt users to add your app to their homescreen, which can lead to higher engagement. [Learn more](#).



## ★ PWA OPTIMIZED

### ● Registers a service worker that controls page and `start_url`

The service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. [Learn more](#).

# Service Worker Cache

```
self.addEventListener('fetch', (event) => {  
  event.respondWith(async function() {  
    try {  
      return await fetch(event.request);  
    } catch (err) {  
      return caches.match(event.request);  
    }  
  })();  
});
```

```
self.addEventListener("install", e => {  
  e.waitUntil(  
    caches.open("static").then(cache => {  
      cache.addAll(['./', './src/styles.css',  
        './images/logo.png', './images/manifest-icon-192.maskable.png',  
        './images/next_song.png', './templates/index.html',  
        './templates/room.html', './templates/create.html',  
        './templates/404.html'])  
    })  
  );  
});
```

You, 1 hour ago • Refactoring code ...

# Service worker push notification

```
messaging.onBackgroundMessage((payload) => {  
  const notificationTitle = payload['notification']['title'];  
  const notificationOptions = {  
    body: payload['notification']['body'],  
    icon: payload['notification']['image']  
  };  
  
  self.registration.showNotification(notificationTitle,  
    notificationOptions);  
});
```



# Push Notification Subscription

```
window.subscribe = async function () {
  getToken(messaging, {vapidKey: "BH9hcXXch_y225Shq9jYBWZVv0AGeDg5hAk_muDhm0tgQEMiLhLQUh5XDHjEfmkpmbj7I
  if (currentToken) {
    console.log(currentToken);
    document.getElementById("pregunta").innerHTML = '';
    fetch('https://ctplmdc.herokuapp.com/api/subscribe', {
      method: 'POST',
      credentials: 'include',
      body: JSON.stringify({
        info: currentToken
      }),
      headers: {
        "accept": "application/json",
        'Content-Type': 'application/json'
      }
    })
  } else {
    console.log('No registration token available. Request permission to generate one.');
```





# Backend manda push notification

```
class SkipSong(APIView):
    def post(self, request, format=None):
        room_code = self.request.session.get('room_code')
        room = Room.objects.filter(code=room_code)[0]
        votes = Vote.objects.filter(room=room, song_id=room.current_song)
        votes_needed = room.votes_to_skip

        if self.request.session.session_key == room.host or len(votes) + 1 >= votes_needed:
            devices = FCMDevice.objects.filter(device_id=room_code)
            devices.send_message(Message(notification=Notification(title="Se ha cambiado la canción",
                                                                    votes.delete()
                                                                    skip_song(room.host)
                                                                    return Response({}, status.HTTP_200_OK)
                                                                    else:
                                                                    vote = Vote(user=self.request.session.session_key,
                                                                    room=room, song_id=room.current_song)
                                                                    vote.save()

                                                                    return Response({}, status.HTTP_204_NO_CONTENT])
```

You, yesterday • Se cambian los modelos ..