

Progressive Web Apps



Grupo 12

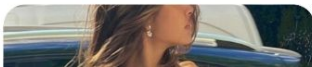


Inicio

Explorar

Crear

Search



Manifesto

```
1  {
2    "name": "Pinterest",
3    "short_name": "Pinterest",
4    "start_url": "/",
5    "display": "standalone",
6    "background_color": "#fdfdfd",
7    "theme_color": "#db4938",
8    "orientation": "portrait-primary",
9    "icons": [
10     {
11       "src": "icons-red/manifest-icon-192.maskable.png",
12       "sizes": "192x192",
13       "type": "image/png",
14       "purpose": "any"
15     },
16     {
17       "src": "icons-red/manifest-icon-192.maskable.png",
18       "sizes": "192x192",
19       "type": "image/png",
20       "purpose": "maskable"
21     },

```

```
npx pwa-asset-generator images/logo.png icons
```

Vista principal

```
4   <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <meta name="theme-color" content="#db4938"/>
8     <title>Pinterest</title>
9     <link rel="stylesheet" href="style/cards.css">
10    <link rel="stylesheet" href="style/navbar.css">
11    <link rel="manifest" href="manifest.json" />
12    <link rel="apple-touch-icon" href="icon-red/manifest-icon-512.maskable.png" />
13    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">
14  </head>
15
16  <body>
17    <div class="pinterest">
18      <div class="left">
19        <a href="#" class="logo">
20          
21        </a>
22      </div>
```

Script de la aplicación

```
108 ✓ const ShowImg = () => {  
109  
110   let output = "";  
111   for (let i = 0; i < page; i++) {  
112     const size = ["card_small", "card_medium", "card_large"].sort(() => Math.random() - 0.5).slice(0, 1);  
113     const element = `https://picsum.photos/id/${i}/200/300`;  
114     output += `  
115               <div class="card ${size}">  
116                 <img src=${element} />  
117               </div>  
118               `;  
119  
120   }  
121   container.innerHTML = output;  
122 }  
123  
124 document.addEventListener("DOMContentLoaded", ShowImg())
```

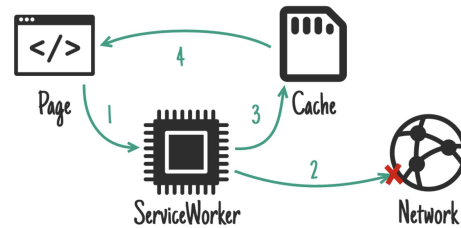
Inicialización del service worker

```
6 // Check if Service Workers are supported
7 if ('serviceWorker' in navigator) {
8   window.addEventListener('load', () => {
9     // Register a Service Worker
10    navigator.serviceWorker.register('/sw.js')
11    .then((registration) => {
12      console.log('Service Worker registration completed with scope: ', registration.scope)
13      // Check if the Push API is supported
14      if ('PushManager' in window) {
15        // Request permission from the user
16        askPermission()
17        .then(() => {
18          const options = {
19            userVisibleOnly: true,
20            applicationServerKey: "BAjli4g8IZem-J1I-ha1zJJJqB_2t-Q3TPVQ9kCFzaHTUN2vUEskKurGhq7XB_rNwRRtKGtrln6tdsjmEgEg-mI"
21          }
22          // Subscribe the user
23          return registration.pushManager.subscribe(options)
24        })
25        // Get the PushSubscription object
26        .then((pushSubscription) => {
27          // Send the PushSubscription object to the server
28          const subscription = JSON.stringify(pushSubscription)
29          sendToServer(subscription)
30        })
31      } else {
32        console.log('Push Api not supported')
33      }
34    }, (err) => {
35      console.log('Service Worker registration failed', err)
36    })
37  })
38 } else {
39   console.log('Service Workers not supported')
40 }
```


Service worker

```
1 self.addEventListener('install', (event) => {
2   console.log('Service worker install event!');
3   const precacheResources = ['/', '/index.html', '/app.js', '/style/cards.css', '/style/navbar.css', '/icons-grey/manifest-icon-512.maskable.png'];
4   event.waitUntil(
5     caches.open('pinterest-cache')
6       .then((cache) => cache.addAll(precacheResources)));
7 });
```

Estrategia de caché



```
13 self.addEventListener('fetch', (event) => {
14   console.log('Fetch intercepted for:', event.request.url);
15   event.respondWith(
16     // Open the cache
17     caches.open('pinterest-cache').then((cache) => {
18       // Go to the network first
19       return fetch(event.request.url, { mode: "cors", cache: "default" }).then((response) => {
20         cache.put(event.request, response.clone());
21         return response;
22       }).catch(() => {
23         // If the network is unavailable, get
24         return caches.match(event.request).then((response) => {
25           if (!response && event.request.destination === 'image') {
26             return caches.match('icons-grey/manifest-icon-192.maskable.png')
27           }
28           return response;
29         });
30       });
31     })
32   );
33 });
```


Eventos push

```
35 // Receive a Push event
36 self.addEventListener('push', (event) => {
37   console.log('Received a push event', event.data.json())
38   const data = event.data.json()
39   console.log(data);
40   const promiseChain = self.registration.showNotification(data.title, data)
41   event.waitUntil(promiseChain)
42 })
```

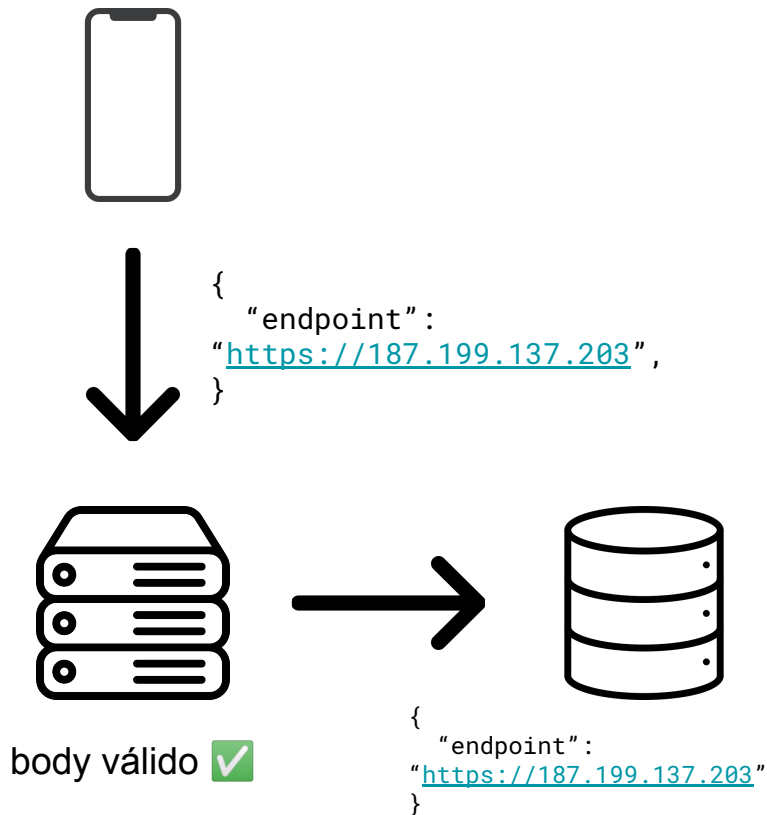
API

- Express
- Firebase hosting
- Firebase functions
- Firestore database / cloud storage



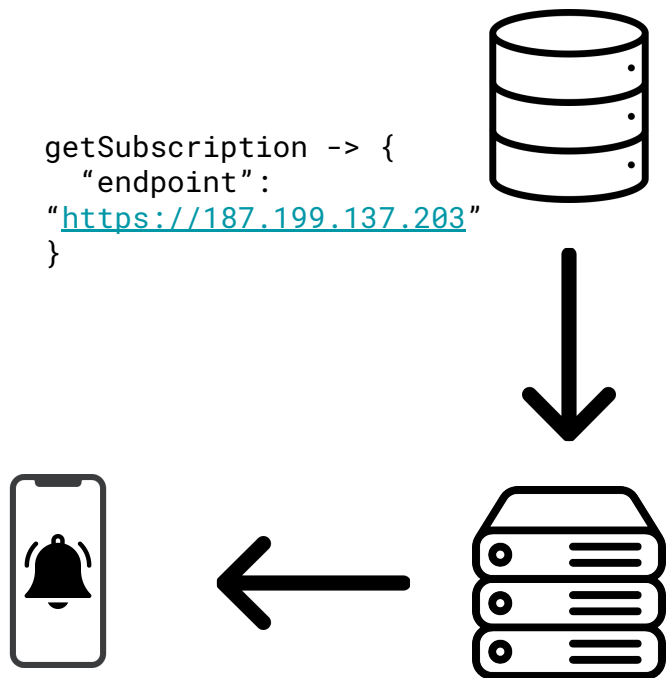
- POST /subscribe
- GET /images

POST /subscription



```
100 app.post('/subscription', async (req, res) => {
101   if (!isValidSaveRequest(req, res)) {
102     return
103   }
104   let subscription = {
105     endpoint: req.body.endpoint
106   }
107   docRef.set(subscription)
108   .then((subscriptionId) => {
109     res.setHeader('Content-Type', 'application/json')
110     res.send(JSON.stringify({ data: { success: true } }))
111   })
112   .catch((err) => {
113     res.status(500)
114     res.setHeader('Content-Type', 'application/json')
115     res.send(JSON.stringify({
116       error: {
117         id: 'unable-to-save-subscription',
118         message: 'Subscription received but failed to save it'
119       }
120     })))
121   })
122   const snapshot = await db.collection('subscriptions').get();
123   snapshot.forEach((doc) => {
124     console.log(doc.id, '=>', doc.data());
125   });
126 })
```

GET /images



```
getSubscription -> {  
  "endpoint":  
  "https://187.199.137.203"  
}
```

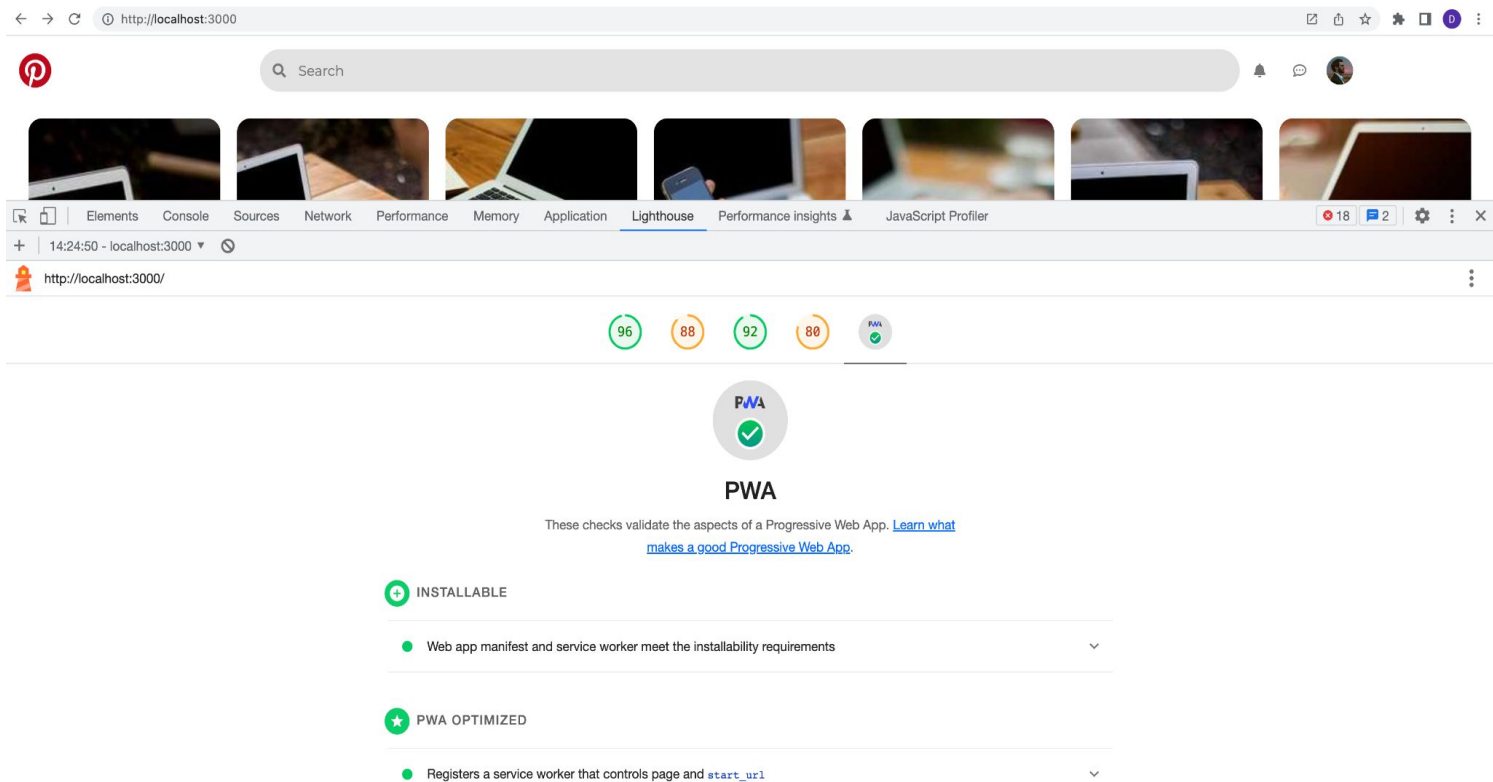
PUSH "<https://187.199.137.203>"

```
129 app.get('/images', (req, res) => {  
130   console.log('Get images')  
131   let dataToSend = {title: '¡Bienvenido/a!', body: 'Esto es lo que te perdiste mientras no estabas.'};  
132   return getSubscriptionsFromDatabase()  
133   .then((subscriptions) => {  
134     let promiseChain = Promise.resolve()  
135     for (let i = 0; i < subscriptions.length; i++) {  
136       const subscription = subscriptions[i]  
137       promiseChain = promiseChain.then(() => {  
138         return triggerPush(subscription, dataToSend)  
139       })  
140     }  
141     return promiseChain  
142   })  
143   .then(() => {  
144     res.setHeader('Content-Type', 'application/json')  
145     res.send(JSON.stringify({ data: dataToSend }))  
146   })  
147   .catch((err) => {  
148     res.status(500)  
149     res.setHeader('Content-Type', 'application/json')  
150     res.send(JSON.stringify({  
151       error: {  
152         id: 'unable-to-send-messages',  
153         message: `Failed to send the push ${err.message}`  
154       }  
155     })))  
156   })  
157 })  
158
```

Envío de notificaciones

```
45 ✓  const triggerPush = (subscription, dataToSend) => {  
46      return webpush.sendNotification(subscription, dataToSend)  
47      .catch((err) => {  
48          if (err.statusCode === 410) {  
49              return deleteSubscriptionFromDatabase(subscription._id)  
50          } else {  
51              console.log('Subscription is no longer valid: ', err)  
52          }  
53      })  
54  }
```

Google Lighthouse



Progressive Web Apps



Grupo 12