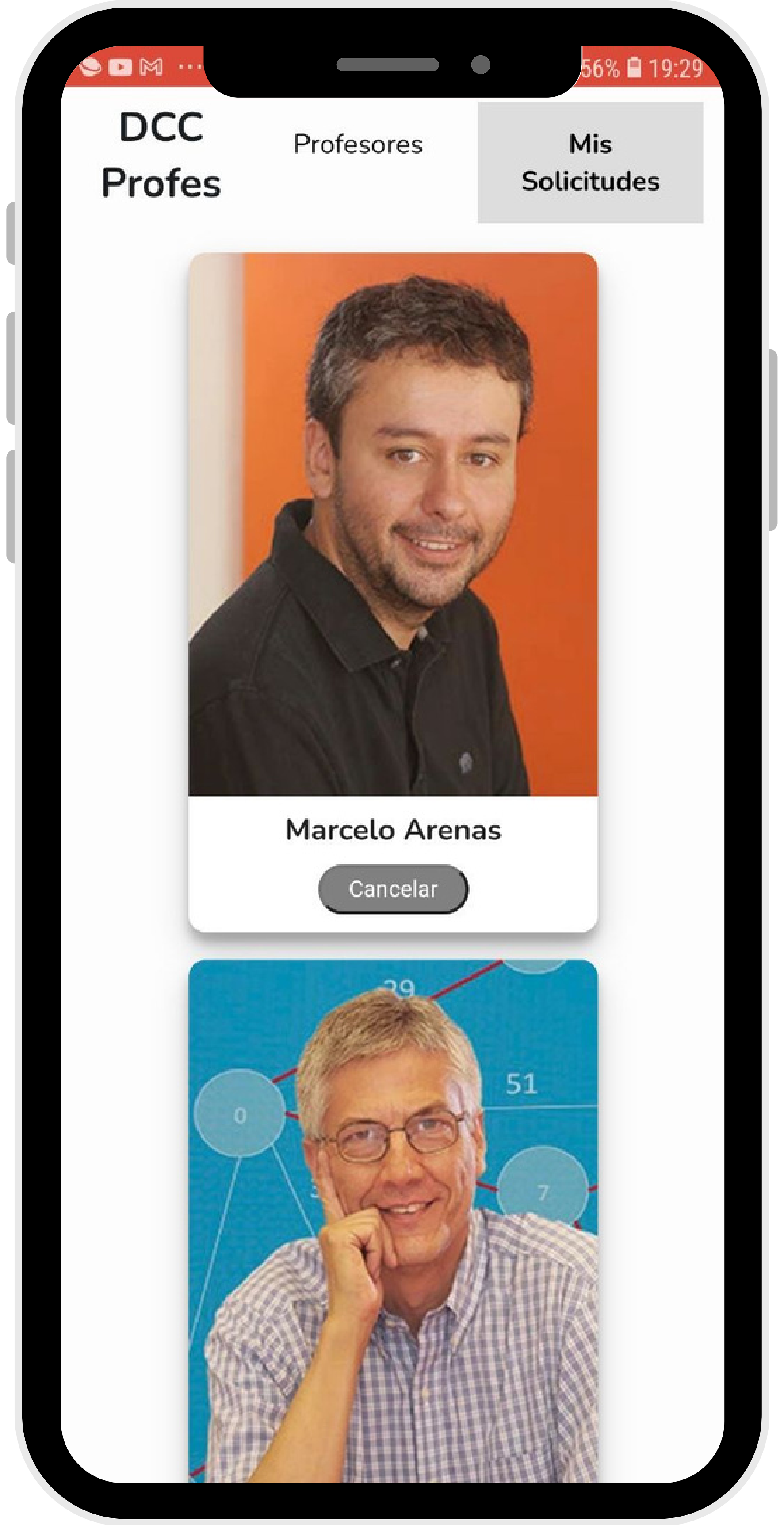
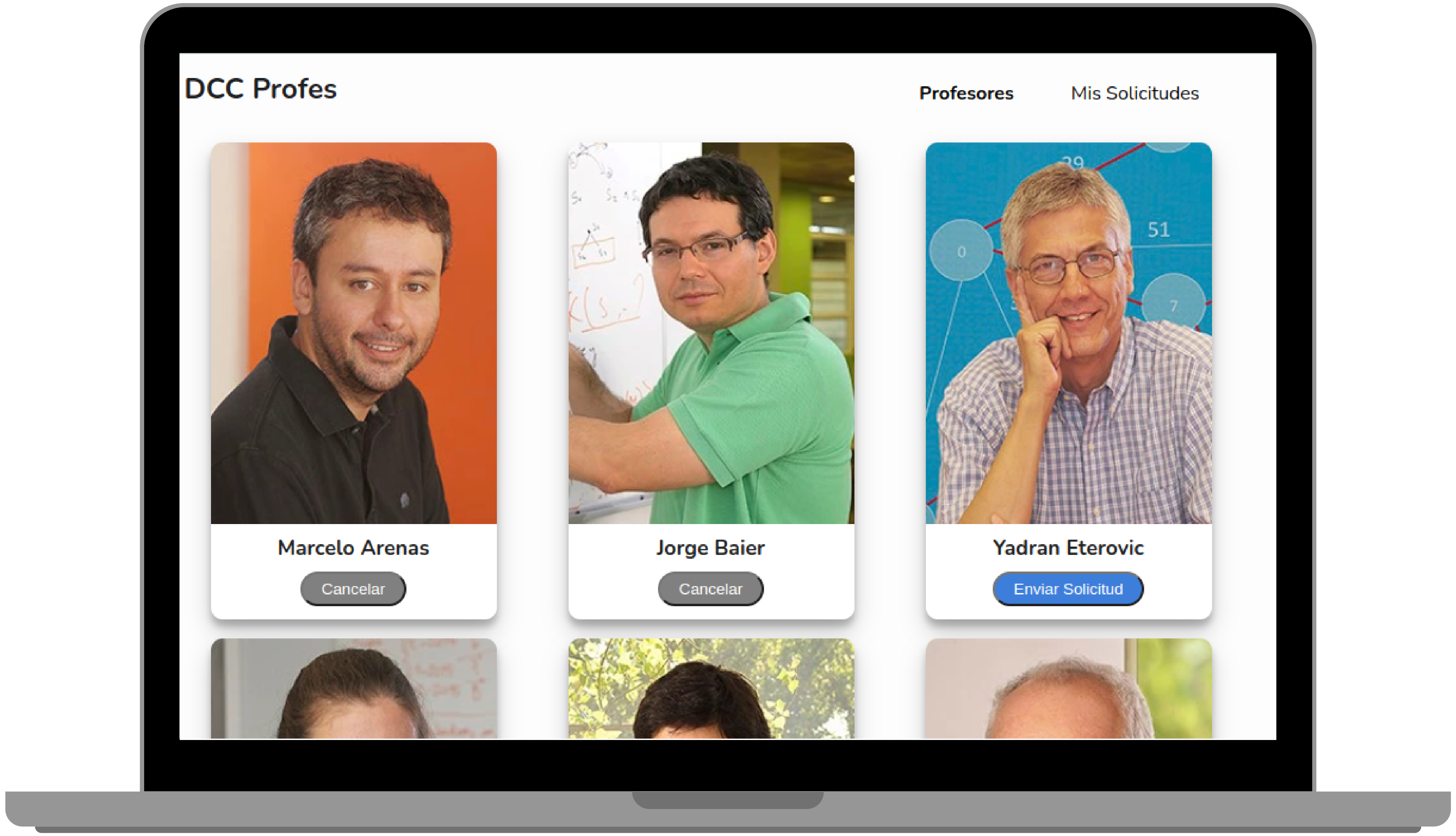
# T4: PWA

Javiera Inostroza, Elías Sabja, Samuel Zúñiga

# DCC Profes


56% 19:29
DCC Profes
Profesores
Mis Solicitudes
Marcelo Arenas
Cancelar
29
51
0
7


DCC Profes
Profesores
Mis Solicitudes
Marcelo Arenas
Cancelar
Jorge Baier
Cancelar
Yadran Eterovic
Enviar Solicitud

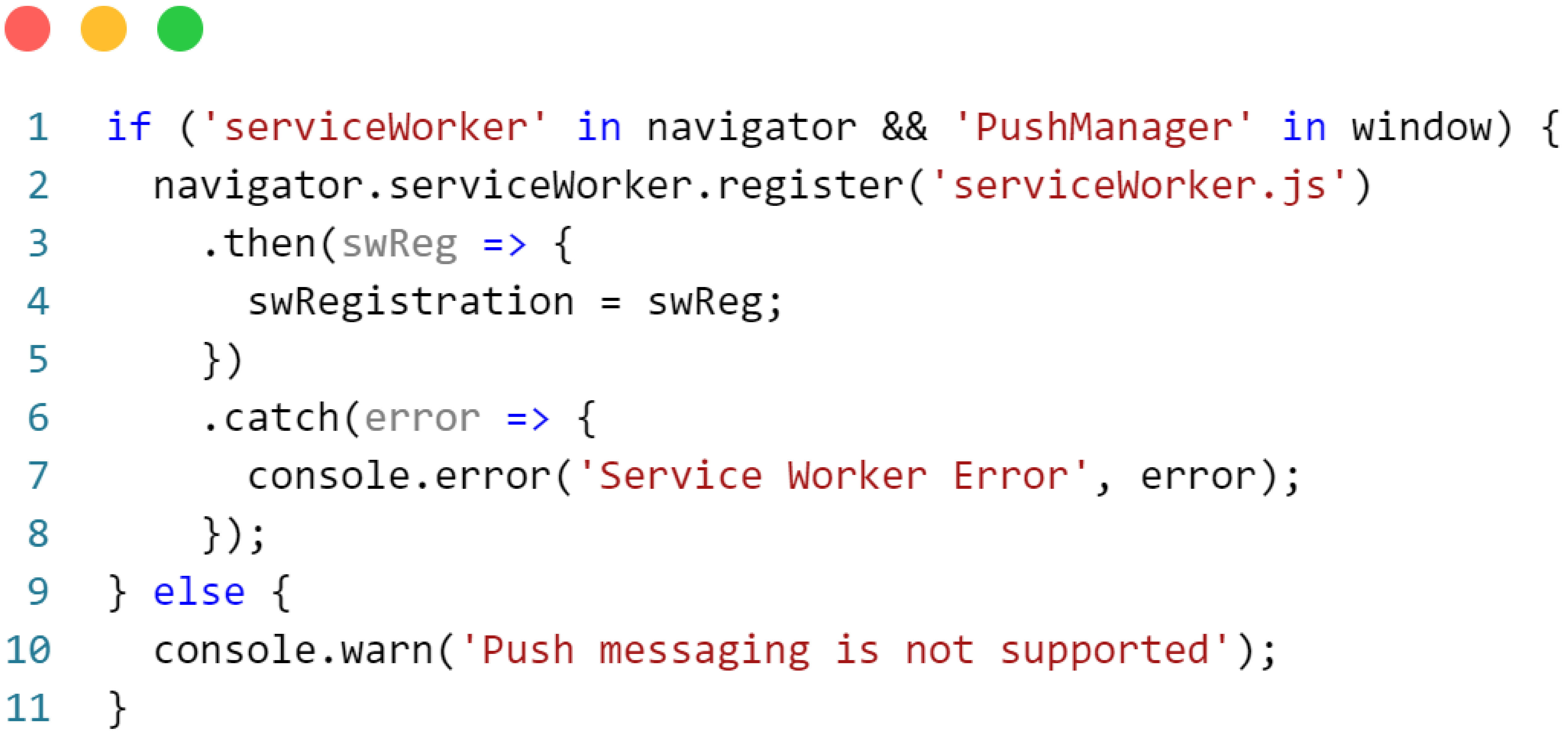
DCC
Profes

# DEMO

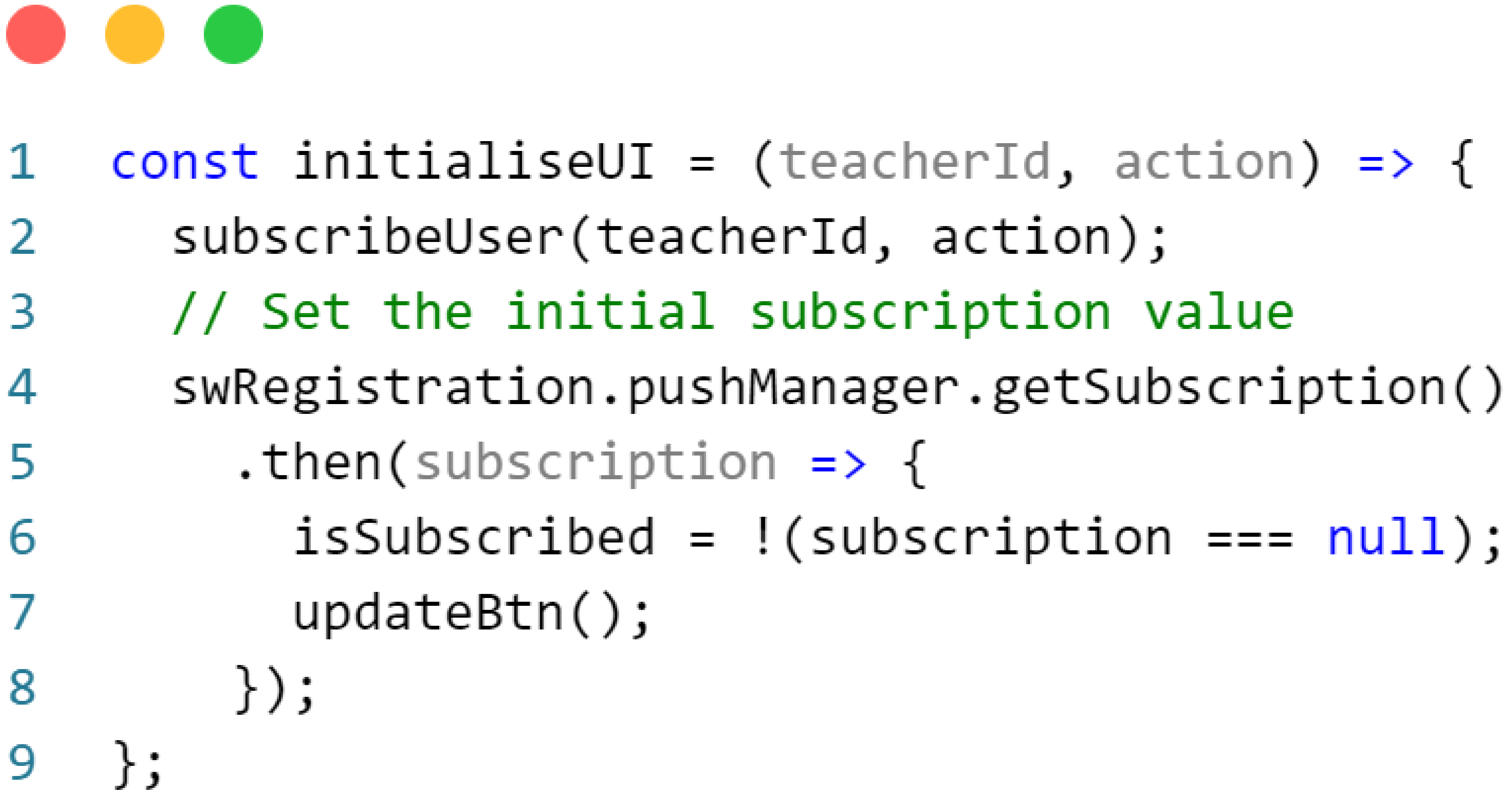
# QR a la App

92% 10:55
Play Store
00:00
eléfono
Gmail
Internet
Cámara

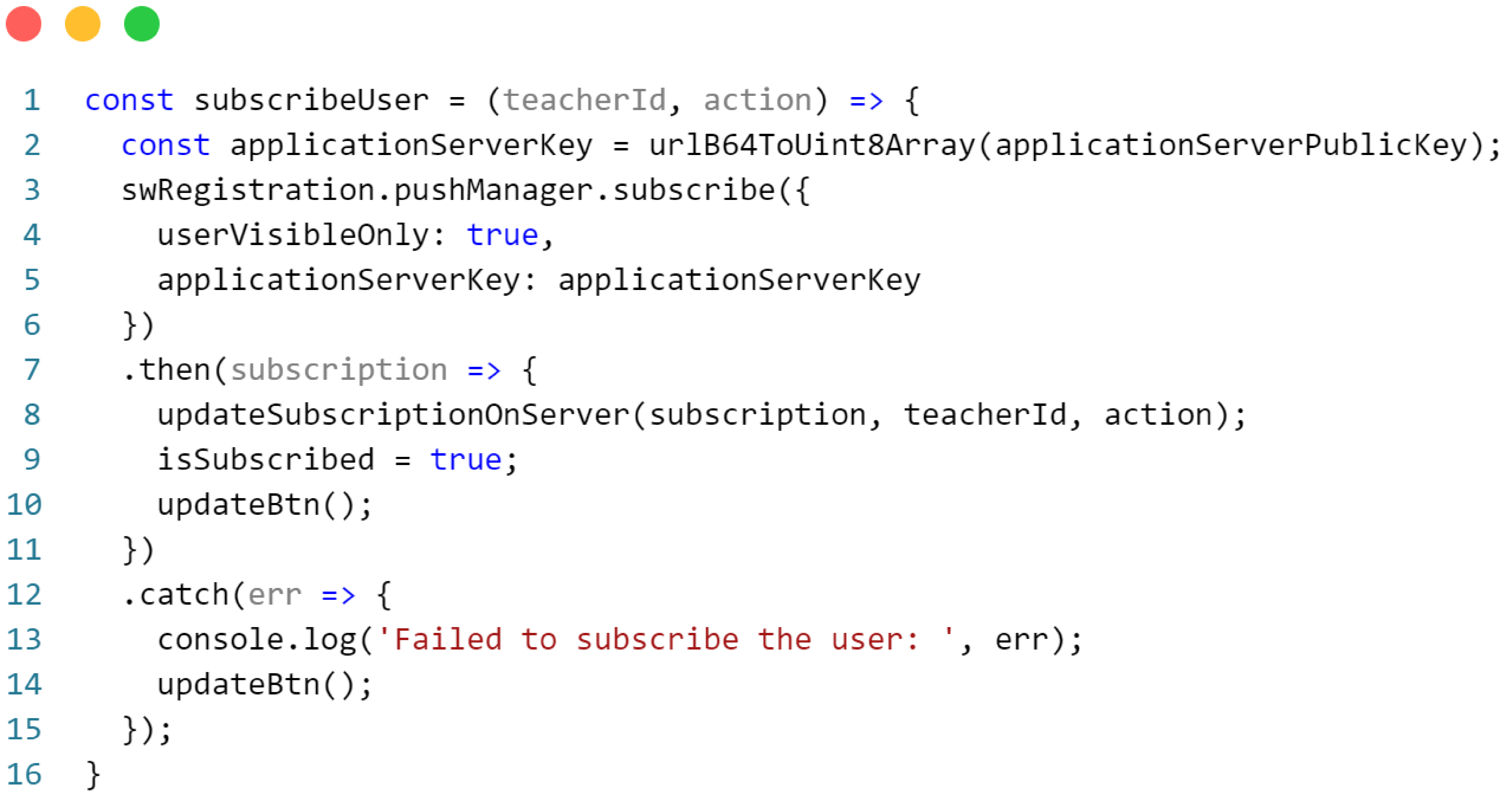
# app.js



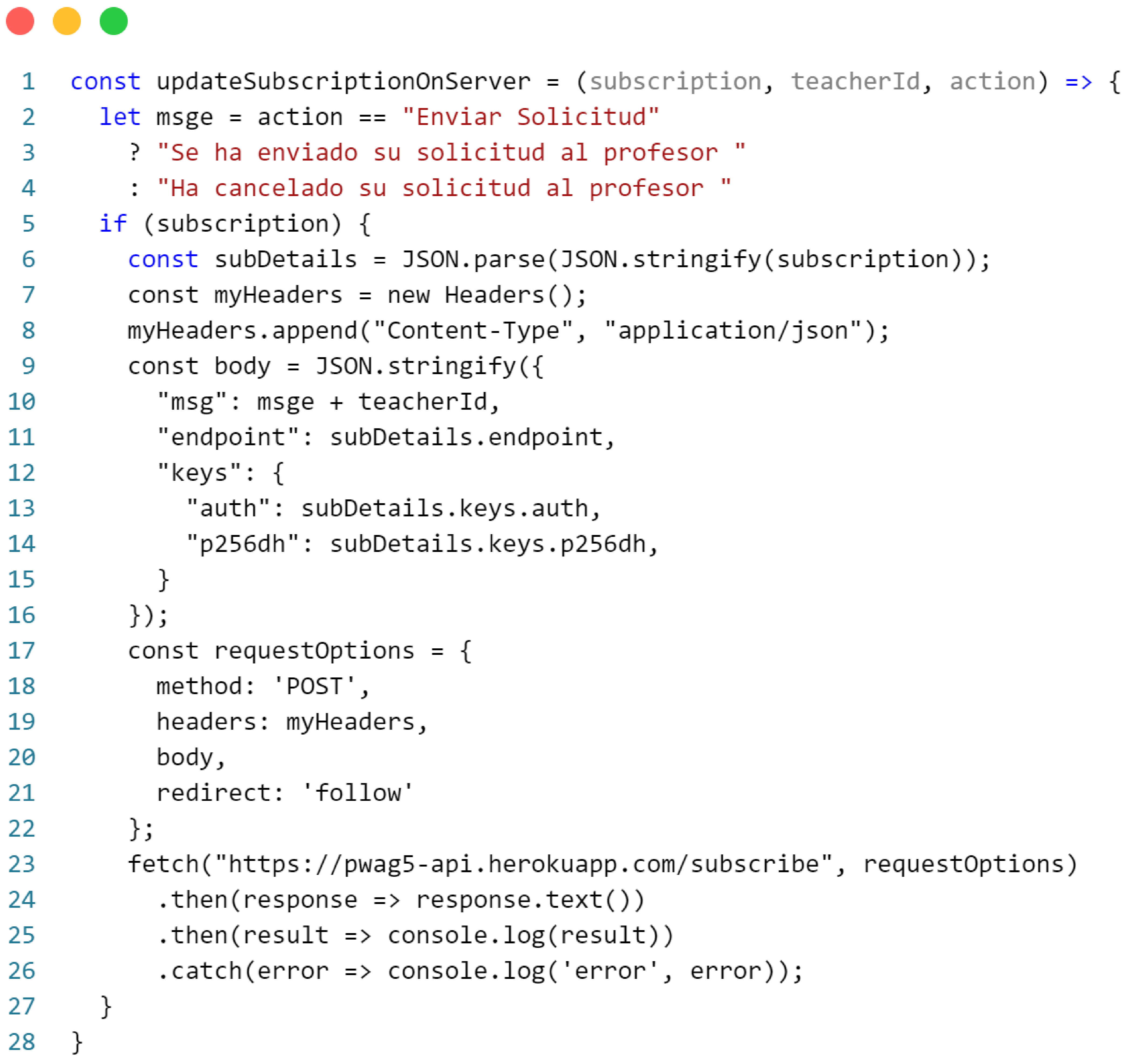
```javascript
if ('serviceWorker' in navigator && 'PushManager' in window) {
  navigator.serviceWorker.register('serviceWorker.js')
    .then(swReg => {
      swRegistration = swReg;
    })
    .catch(error => {
      console.error('Service Worker Error', error);
    });
} else {
  console.warn('Push messaging is not supported');
}
```



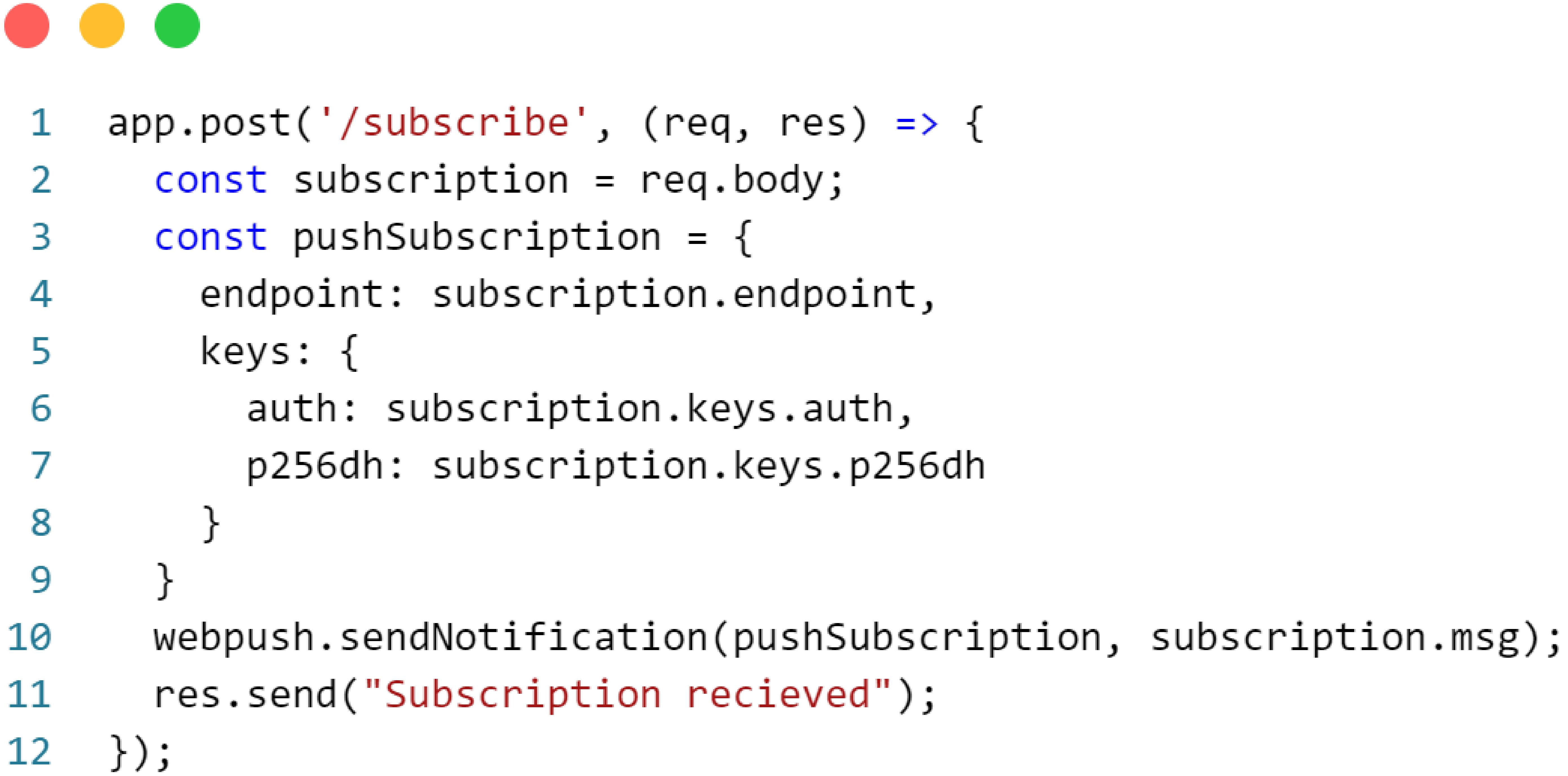
```
const initialiseUI = (teacherId, action) => {
  subscribeUser(teacherId, action);
  // Set the initial subscription value
  swRegistration.pushManager.getSubscription()
    .then(subscription => {
      isSubscribed = !(subscription === null);
      updateBtn();
    });
};
```



```javascript
const subscribeUser = (teacherId, action) => {
  const applicationServerKey = urlB64ToUint8Array(applicationServerPublicKey);
  swRegistration.pushManager.subscribe({
    userVisibleOnly: true,
    applicationServerKey: applicationServerKey
  })
  .then(subscription => {
    updateSubscriptionOnServer(subscription, teacherId, action);
    isSubscribed = true;
    updateBtn();
  })
  .catch(err => {
    console.log('Failed to subscribe the user: ', err);
    updateBtn();
  });
}
```

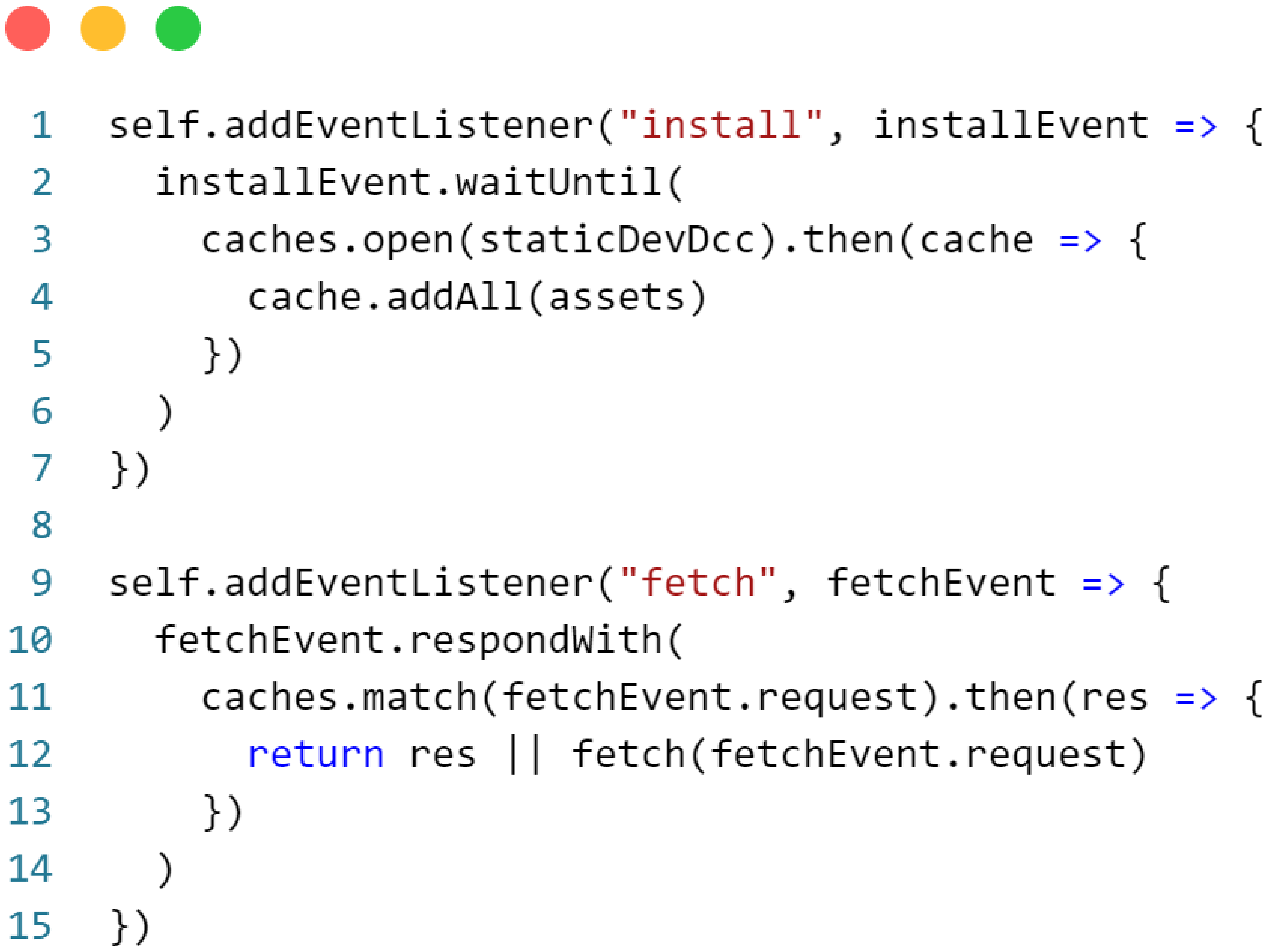


```
const updateSubscriptionOnServer = (subscription, teacherId, action) => {
  let msge = action == "Enviar Solicitud"
    ? "Se ha enviado su solicitud al profesor "
    : "Ha cancelado su solicitud al profesor "
  if (subscription) {
    const subDetails = JSON.parse(JSON.stringify(subscription));
    const myHeaders = new Headers();
    myHeaders.append("Content-Type", "application/json");
    const body = JSON.stringify({
      "msg": msge + teacherId,
      "endpoint": subDetails.endpoint,
      "keys": {
        "auth": subDetails.keys.auth,
        "p256dh": subDetails.keys.p256dh,
      }
    });
    const requestOptions = {
      method: 'POST',
      headers: myHeaders,
      body,
      redirect: 'follow'
    };
    fetch("https://pwag5-api.herokuapp.com/subscribe", requestOptions)
      .then(response => response.text())
      .then(result => console.log(result))
      .catch(error => console.log('error', error));
  }
}
```

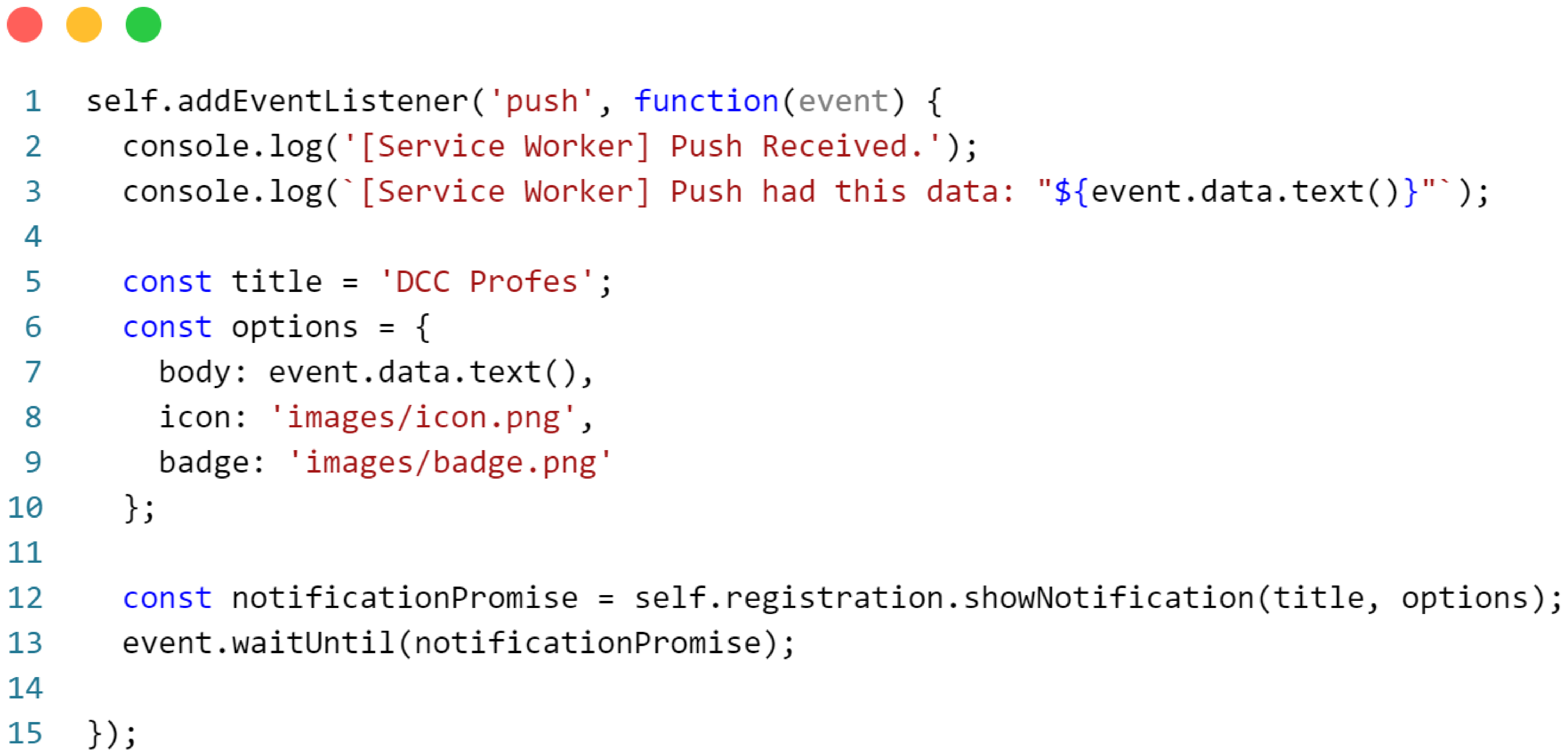


```javascript
app.post('/subscribe', (req, res) => {
  const subscription = req.body;
  const pushSubscription = {
    endpoint: subscription.endpoint,
    keys: {
      auth: subscription.keys.auth,
      p256dh: subscription.keys.p256dh
    }
  }
  webpush.sendNotification(pushSubscription, subscription.msg);
  res.send("Subscription recieved");
});
```

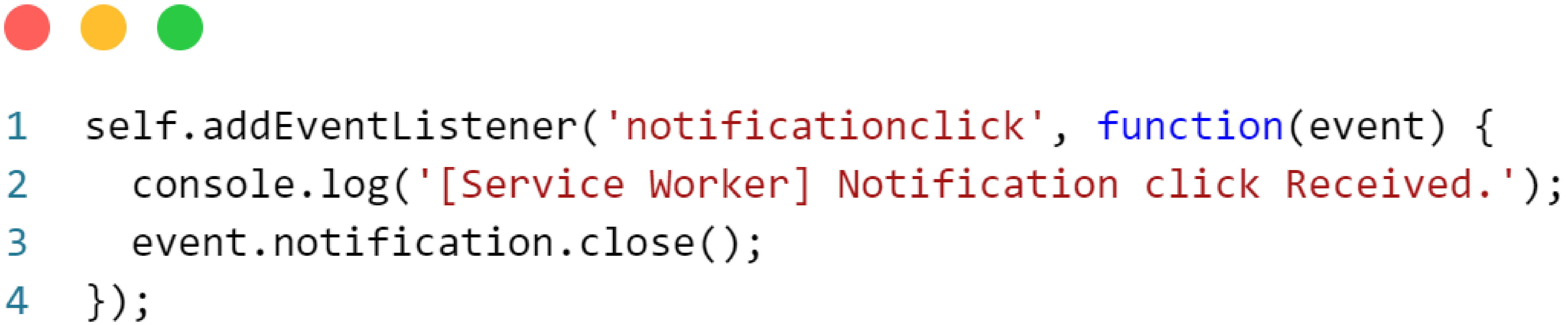
# serviceWorker.js



```javascript
self.addEventListener("install", installEvent => {
  installEvent.waitUntil(
    caches.open(staticDevDcc).then(cache => {
      cache.addAll(assets)
    })
  )
})

self.addEventListener("fetch", fetchEvent => {
  fetchEvent.respondWith(
    caches.match(fetchEvent.request).then(res => {
      return res || fetch(fetchEvent.request)
    })
  )
})
```

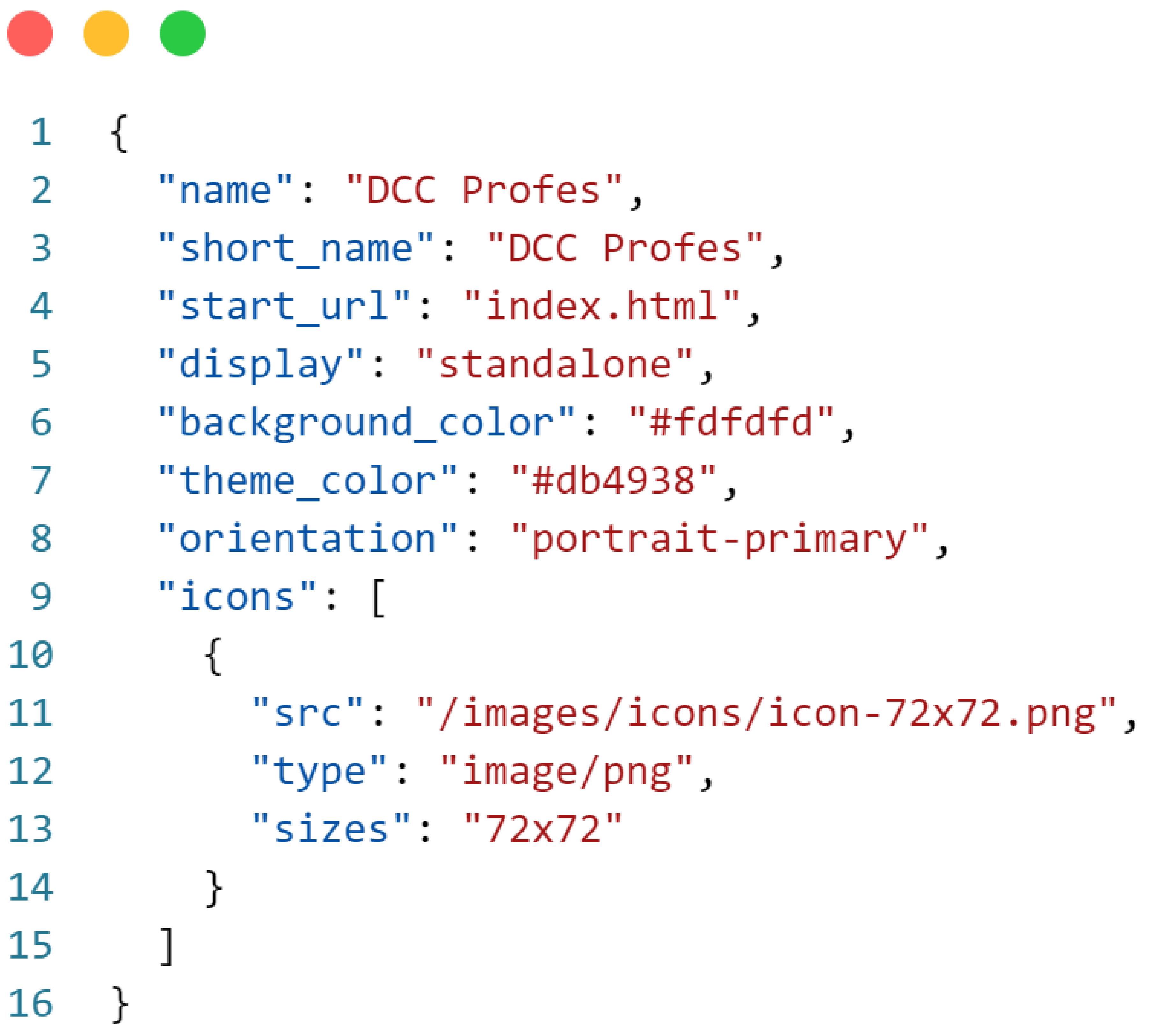


```javascript
self.addEventListener('push', function(event) {
  console.log('[Service Worker] Push Received.');
  console.log(`[Service Worker] Push had this data: "${event.data.text()}"`);

  const title = 'DCC Profes';
  const options = {
    body: event.data.text(),
    icon: 'images/icon.png',
    badge: 'images/badge.png'
  };

  const notificationPromise = self.registration.showNotification(title, options);
  event.waitUntil(notificationPromise);

});
```



```javascript
self.addEventListener('notificationclick', function(event) {
  console.log('[Service Worker] Notification click Received.');
  event.notification.close();
});
```

# manifest.json



```json
{
  "name": "DCC Profes",
  "short_name": "DCC Profes",
  "start_url": "index.html",
  "display": "standalone",
  "background_color": "#fdfdfd",
  "theme_color": "#db4938",
  "orientation": "portrait-primary",
  "icons": [
    {
      "src": "/images/icons/icon-72x72.png",
      "type": "image/png",
      "sizes": "72x72"
    }
  ]
}
```

# DIFICULTADES Y CONCLUSIONES

Implementar notificaciones push

Problemas de hacer fetch para las notificaciones push

Guardar caché es poco práctico para debuggear

Sigue existiendo el problema de que la gente llegue a la aplicación

Muy útil poder cachear datos que funcionen de manera offline

Práctico poder guardar una aplicación web como ícono de app

# T4: PWA

Javiera Inostroza, Elías Sabja, Samuel Zúñiga