

PWA

La resurrección del chat de Terra 🐒

Jorge Araneda

Vicente Calisto

Tristan Heuer



Hace ~20 años atrás...





Hoy



CURIOSIDADES

Alonso "Monito" Vidal: 5 curiosidades del hijo mayor de Arturo Vidal que no conocías

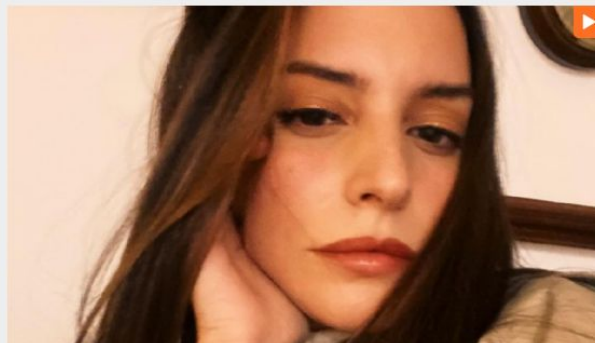
Alonso cumplió 13 años el pasado 7 de mayo, y todos lo saludaron en sus redes.



SECRETOS

Esta es la mujer que rechazó a Chayanne pero aceptó a Luis Miguel

ENTRETENIMIENTO



INTIMIDADES

La verdad detrás del romance de la hija de José Luis 'el Puma' Rodríguez y Paul Walker

ENTRETENIMIENTO

¡Revivamos el **Chat de Terra!**

Ingresen a

<https://pwa-grupo8.tk/>

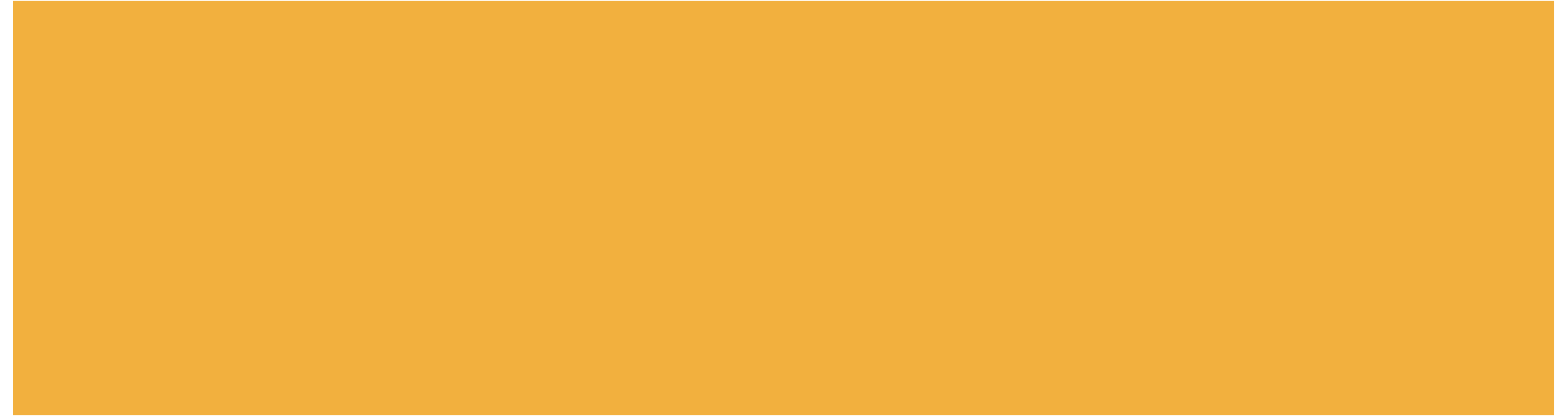
Frontend

Manifest



```
1  {
2    "name": "Chat de Terra",
3    "start_url": "/",
4    "display": "standalone",
5    "id": "grupo-8-pwa",
6    "background_color": "#FFFFFF",
7    "theme_color": "#ffa200",
8    "icons": [
9      {
10         "src": "/favicon.png",
11         "type": "image/png",
12         "purpose": "maskable any",
13         "sizes": "any"
14       }
15     ]
16  }
```

Service Worker: Registro



```
26  if ('serviceWorker' in navigator) {  
27      navigator.serviceWorker  
28          .register('/service-worker.js')  
29          .then(function (registration) {  
30              console.log(  
31                  'Registration successful, scope is:',  
32                  registration.scope  
33              );  
34          })  
35          .catch(function (error) {  
36              console.log('Service worker registration failed, error:', error);  
37          });  
38      }
```

Service Worker: Cache



```
21 self.addEventListener('install', async (event) => {
22     console.log('installing!');
23     const cache = await self.caches.open(CACHE_NAME);
24     await cache.addAll(urlsToCache);
25     self.skipWaiting();
26 });
```

```
28 self.addEventListener('activate', async (event) => {
29     console.log('activating!');
30     const keys = await caches.keys();
31     keys
32         .filter((key) => key !== CACHE_NAME)
33         .map((key) => event.waitUntil(caches.delete(key)));
34 });
35
```

```
36 async function networkFirst(request) {
37   const cache = await self.caches.open(CACHE_NAME);
38   try {
39     const response = await fetch(request);
40     cache.put(request, response.clone());
41     console.log('Network: ', request.url);
42     return response;
43   } catch {
44     console.log('Cache: ', request.url);
45     return cache.match(request);
46   }
47 }
48
49 self.addEventListener('fetch', (event) => {
50   if (event.request.method !== 'GET') return;
51   event.respondWith(networkFirst(event.request));
52 });
```


Token: Identificador de Instancia



```
64 getToken(messaging, { vapidKey })
65   .then((currentToken) => {
66     if (currentToken) {
67       console.log(currentToken);
68       sendRequest('/token', 'POST', { token: currentToken }).then(
69         (response) => console.log('Send token response:', response)
70       );
71     } else {
72       // Show permission request UI
73       console.log(
74         'No registration token available. Request permission to generate one.'
75       );
76     }
77   })
78   .catch((err) => {
79     console.log('An error occurred while retrieving token. ', err);
80   });
```

Notificaciones Push: Recepción

Firebase Cloud Messaging (FCM)

```
19 messaging.onBackgroundMessage((payload) => {
20   console.log('Background: ', payload.data);
21
22   console.log('Sending data to notification');
23   try {
24     self.clients.matchAll({ includeUncontrolled: true }).then((clients) => {
25       clients.forEach((client) => {
26         console.log('Sending to client ' + client);
27         client.postMessage({
28           data: payload.data,
29         });
30       });
31     });
32   } catch (e) {
33     console.log(e);
34   }
35
36   // Customize notification here
37   const notificationTitle = '¡Recibiste un nuevo mensaje!';
38   const notificationOptions = {
39     body: 'Ve al Chat de Terra para revisarlo',
40     icon: '/favicon.png',
41   };
42
43   self.registration.showNotification(notificationTitle, notificationOptions);
44 });
```

Backend

Notificaciones Push: Envío

Firebase Cloud Messaging (FCM)

```
30 func NewService(logger *log.Logger) (service, error) {
31     opt := option.WithCredentialsJSON(config.FIREBASE)
32     app, err := firebase.NewApp(context.Background(), nil, opt)
33     if err != nil {
34         return service{}, fmt.Errorf("error initializing app: %v", err)
35     }
36
37     msgClient, err := app.Messaging(context.Background())
38     if err != nil {
39         return service{}, fmt.Errorf("error initializing app: %v", err)
40     }
41
42     return service{app, msgClient, logger}, nil
43 }
```

```
45 func (s service) SubmitToken(ctx context.Context, token string) (*types.Error) {
46     resp, err := s.msgClient.SubscribeToTopic(ctx, []string{token}, "chat")
47
48     for _, e := range resp.Errors {
49         s.logger.Debug("%v", e.Reason)
50     }
51
52     s.logger.Info("Subscription: success: %v, failure: %v\n", resp.SuccessCount, resp.FailureCount)
53
54     if err != nil {
55         s.logger.Critical(err)
56         return types.UnexpectedError
57     }
58
59     return nil
60 }
```



```
62 func (s service) SendMessage(ctx context.Context, user string, content string) (*types.Error) {
63     message := &messaging.Message{
64         Data: map[string]string{
65             "user": user,
66             "content": content,
67         },
68         Topic: "chat",
69     }
70     // Send a message to the devices subscribed to the provided topic.
71     response, err := s.msgClient.Send(ctx, message)
72
73     s.logger.Debug("SEND: %v\n", response)
74     if err != nil {
75         s.logger.Critical(err)
76         return types.UnexpectedError
77     }
78     return nil
79 }
```

Conclusiones

Aplicación instalable

- Es muy fácil llegar a una aplicación instalable
- No es necesario usar app stores
- Es fácil de distribuir e instalar
- Funciona en todos los dispositivos

Features de aplicaciones nativas

- Push messages
- Notificaciones
- Offline