



ANGULAR & FIREBASE

CREATING A CHAT ROOM
IN REAL TIME

By Alejandro Lora

AGENDA

SPONSOR

INTRODUCCIÓN

CREANDO PROYECTO

LOGIN BY GOOGLE

MENSAJES

NOTIFICACIONES

DEPLOYMENT

DEMO LIVE

ALEJANDRO LORA

- Freelance Software Developer enfocado en Angular.
- Instructor en Udemy con +28K alumnos.
- Reviewer en “Learning Typescript 2.X: second edition”.
- Sígueme en alejandrolora.com,
[@alexpc](https://twitter.com/alexpc) o github.com/ialex90.





SPONSOR

PACKT PUB

LIBRO DIARIO GRATIS: <https://www.packtpub.com/packt/offers/free-learning>

OFERTA DIARIAS: <https://www.packtpub.com/booksdeal-of-the-day>

MAPT: <https://mapt.packtpub.com/>

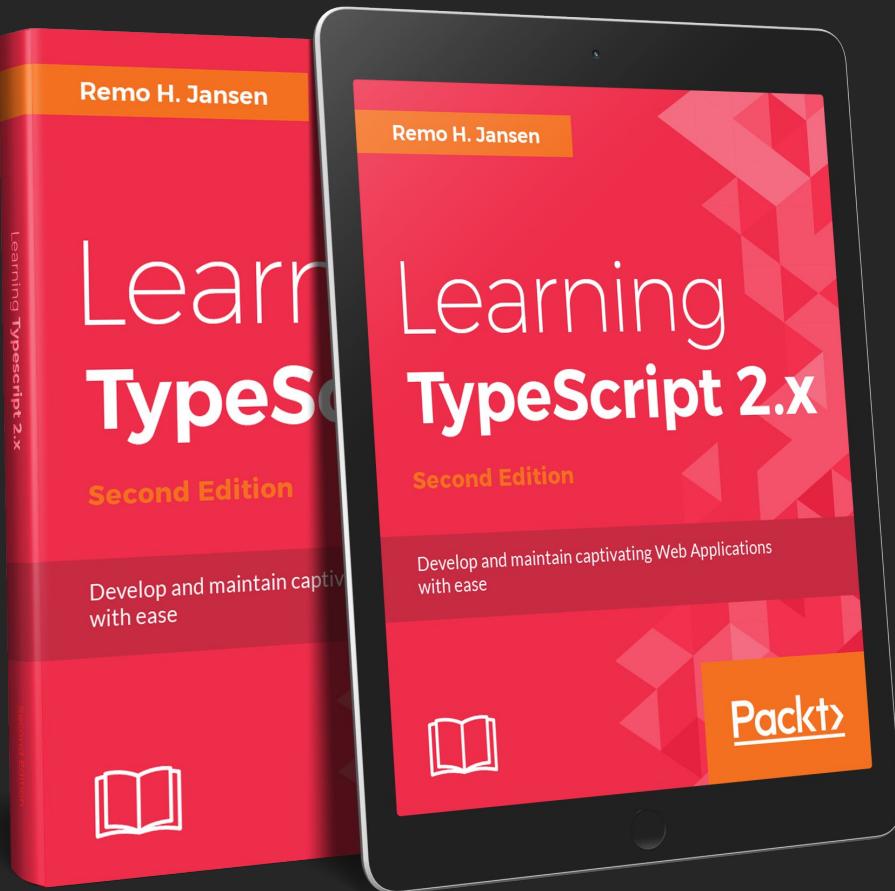
SUBSCRIPCIÓN MAPT MUY COMPLETA:

- +6.000 LIBROS Y VIDEOS.
- +100 NUEVOS TÍTULOS MENSUALMENTE.
- OBTEN 1 VÍDEO O LIBRO OFFLINE CADA MES GRATIS.
- PERIODO PRUEBA 14 DÍAS

PACKT PUB

SORTEO DE 2 EBOOKS GRATIS
PATROCINADO POR PACKT PUB

[https://www.amazon.es/Learning-TypeScript-2-x Second Edition/dp/1788391470/](https://www.amazon.es/Learning-TypeScript-2-x-Second-Edition/dp/1788391470/)





INTRODUCCIÓN

SOBRE ANGULAR

AngularJS fué la versión anterior de Angular, el cual fue escrito desde cero.

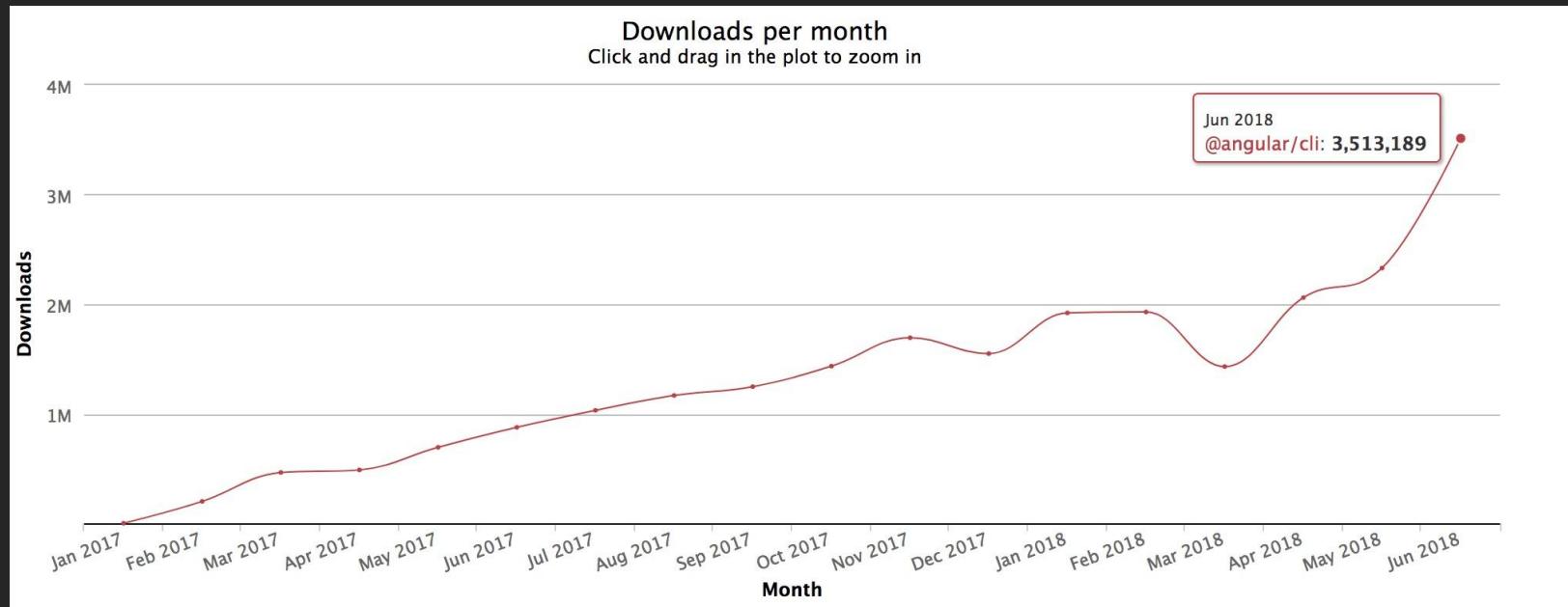
No es necesario diferenciar versiones de Angular como no lo es para otras librerías/frameworks (ReactJS 16, Vue 2, Nativescript 4...)

Versión 6 publicada recientemente. Patch releases casi semanales, Minor releases cada 2 meses approx y Major releases cada 6 meses.

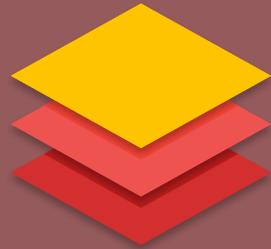
Mobile Apps, Desktop Apps, Web Apps.

DATE	STABLE RELEASE	COMPATIBILITY
September/October 2018	7.0.0	[^] 6.0.0
March/April 2019	8.0.0	[^] 7.0.0

ANGULAR CLI - NPM



<https://npm-stat.com/charts.html?package=%40angular%2Fcli&from=2017-01-01&to=2018-06-30>



CREANDO PROYECTO

ANGULAR CLI

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
> npm install -g @angular/cli  
> ng new my-dream-app  
> cd my-dream-app  
> ng serve
```

```
ng new my-app --style=scss --routing
```

* Se requiere previa instalación de Node y VSCode.

ESTRUCTURA

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
▶ 📁 .git
▶ 📁 e2e
▶ 📁 node_modules
▶ 📁 src —————→
  🎨 .editorconfig
  🚫 .gitignore
  🏙 angular.json
  📜 package-lock.json
  📜 package.json
  📄 README.md
  📜 tsconfig.json
  📜 tslint.json
```

```
◀ 📁 app
  📜 app-routing.module.ts
  📜 app.component.html
  📜 app.component.scss
  📜 app.component.spec.ts
  🏙 app.component.ts
  📜 app.module.ts
  📁 assets
  📁 environments
  📜 browserslist
  🇮 favicon.ico
  📜 index.html
  📜 karma.conf.js
  📜 main.ts
  📜 polyfills.ts
  📜 styles.scss
  📜 test.ts
  📜 tsconfig.app.json
  📜 tsconfig.spec.json
  📜 tslint.json
```

ANGULAR MATERIAL

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
npm install --save @angular/material @angular/cdk
```

```
npm install --save @angular/animations
```

```
npm install --save hammerjs
```

<https://material.angular.io/guide/getting-started>

MATERIAL MODULE

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
material.module.ts ✘
1   import { NgModule } from '@angular/core';
2   import { MatToolbarModule } from '@angular/material/toolbar';
3   import { MatButtonModule } from '@angular/material/button';
4   import { MatMenuModule } from '@angular/material/menu';
5   import { MatIconModule } from '@angular/material/icon';
6   import { MatProgressBarModule } from '@angular/material/progress-bar';
7   import { MatFormFieldModule } from '@angular/material/form-field';
8   import { MatInputModule } from '@angular/material/input';
9
10  @NgModule({
11    imports: [
12      MatToolbarModule,
13      MatButtonModule,
14      MatMenuModule,
15      MatIconModule,
16      MatProgressBarModule,
17      MatFormFieldModule,
18      MatInputModule
19    ],
20    exports: [
21      MatToolbarModule,
22      MatButtonModule,
23      MatMenuModule,
24      MatIconModule,
25      MatProgressBarModule,
26      MatFormFieldModule,
27      MatInputModule
28    ]
29  })
30  export class MaterialModule { }
```

THEMING

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)**my-theme.scss**

```
1  @import '~@angular/material/theming';
2  @include mat-core();
3
4  $primary: mat-palette($mat-deep-purple);
5  $accent: mat-palette($mat-amber);
6  $warn: mat-palette($mat-red);
7  $theme: mat-light-theme($primary, $accent, $warn);
8
9  @include angular-material-theme($theme);
```

styles.scss

```
1  /* You can add global styles to this file */
2  @import 'my-theme.scss';
3
4  html, html body {
5    margin: 0;
6  }
7
8  .spacer {
9    flex-grow: 1;
10 }
```

HOME COMPONENT

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
mbpdealejandro:angular-firebase-gdg alejandrolora$ ng g module home
CREATE src/app/home/home.module.spec.ts (259 bytes)
CREATE src/app/home/home.module.ts (188 bytes)
mbpdealejandro:angular-firebase-gdg alejandrolora$ ng g component home
CREATE src/app/home/home.component.scss (0 bytes)
CREATE src/app/home/home.component.html (23 bytes)
CREATE src/app/home/home.component.spec.ts (614 bytes)
CREATE src/app/home/home.component.ts (262 bytes)
UPDATE src/app/home/home.module.ts (251 bytes)
```

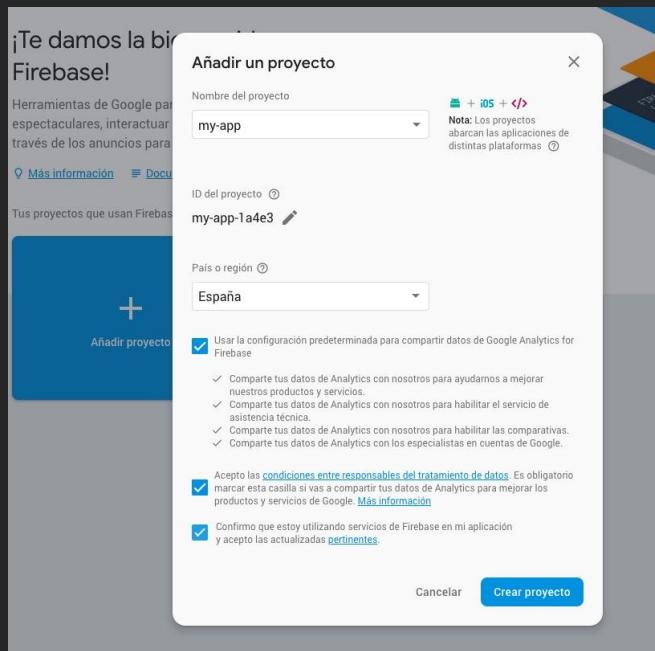
[home.component.ts x](#)

```
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-home',
5   templateUrl: './home.component.html',
6   styleUrls: ['./home.component.scss']
7 })
8 export class HomeComponent implements OnInit {
9
10   isAuthenticated = false;
11
12   constructor() { }
13
14   ngOnInit() { }
15
16   login() {
17     this.isAuthenticated = true;
18   }
19
20   logout() {
21     this.isAuthenticated = false;
22   }
23 }
```

[home.component.html x](#)

```
1 <mat-toolbar color="primary" class="mat-elevation-z6">
2   <span>Angular & Firebase</span>
3   <div class="spacer"></div>
4   <button *ngIf="!isAuthenticated" mat-raised-button color="accent" (click)="login()">Log In</button>
5   <button *ngIf="isAuthenticated" mat-raised-button color="accent" (click)="logout()">Log Out</button>
6 </mat-toolbar>
```

PROYECTO EN FIREBASE

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

Añade Firebase a tu aplicación web

Copia y pega el fragmento que se indica a continuación en la parte inferior de tu código HTML, delante de otras etiquetas de secuencia de comandos.

```
<script src="https://www.gstatic.com/firebasejs/5.2.0.firebaseio.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "AIzaSyCnTxp73fAQbWG86sSAHXAwc3uVINrcNYs",
    authDomain: "my-app-1a4e3.firebaseioapp.com",
    databaseURL: "https://my-app-1a4e3.firebaseio.com",
    projectId: "my-app-1a4e3",
    storageBucket: "",
    messagingSenderId: "123928764926"
  };
  firebase.initializeApp(config);
</script>
```

[Copiar](#)

ANGULARFIRE2

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)**TS environment.ts**

```
1  export const environment = {  
2    production: false,  
3    firebase: {  
4      apiKey: 'AIzaSyAPil2za_u2I-a8aln44xFv33djjxSLVss',  
5      authDomain: 'angularfirebasegdg.firebaseio.com',  
6      databaseURL: 'https://angularfirebasegdg.firebaseio.com',  
7      projectId: 'angularfirebasegdg',  
8      storageBucket: 'angularfirebasegdg.appspot.com',  
9      messagingSenderId: '26732418254'  
10    }  
11  };
```

A app.module.ts

```
1  import { BrowserModule } from '@angular/platform-browser';  
2  import { NgModule } from '@angular/core';  
3  import { AngularFireModule } from 'angularfire2';  
4  import { AngularFirestoreModule } from 'angularfire2/firestore';  
5  import { AngularFireAuthModule } from 'angularfire2/auth';  
6  
7  import { AppComponent } from './app.component';  
8  import { HomeModule } from './home/home.module';  
9  import { environment } from '../environments/environment';  
10  
11 @NgModule({  
12   imports: [  
13     BrowserModule,  
14     AngularFireModule.initializeApp(environment.firebaseio),  
15     AngularFirestoreModule,  
16     AngularFireAuthModule,  
17     HomeModule  
18   ],  
19   declarations: [AppComponent],  
20   bootstrap: [AppComponent]  
21 })  
22 export class AppModule { }
```



LOGIN BY
GOOGLE

USER INTERFACE

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

TS user.ts 

```
1  export interface User {  
2      uid: string;  
3      displayName: string;  
4      email: string;  
5      photoURL: string;  
6  }
```

AUTH SERVICE

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
auth.service.ts x
1 import { Injectable } from '@angular/core';
2 import { Subject } from 'rxjs';
3
4 import { auth } from 'firebase/app';
5 import { AngularFireAuth } from 'angularfire2/auth';
6 import { User } from './user';
7
8 @Injectable({ providedIn: 'root' })
9 export class AuthService {
10
11   private currentUserSource = new Subject<User>();
12   public currentUser$ = this.currentUserSource.asObservable();
13
14   constructor(private afAuth: AngularFireAuth) {
15     this.syncCurrentUser();
16   }
17
18   loginByGoogle() {
19     const provider = new auth.GoogleAuthProvider();
20     return this.afAuth.auth.signInWithPopup(provider);
21   }
22
23   logout() {
24     this.afAuth.auth.signOut();
25   }
26
27   userState() {
28     return this.afAuth.authState;
29   }
}
```

Exponiendo currentUser como observable.

Abre Google Sign-In PopUp.

AUTH SERVICE

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
A auth.service.ts x
30
31 private syncCurrentUser() {
32   this.afAuth.authState.subscribe((user) => {
33     const result = (user) ? this.getCurrentUser(user) : null;
34     this.currentUserSource.next(result);
35   });
36 }
37
38 private getCurrentUser(user: any) {
39   const currentUser = <User>{};
40   currentUser.uid = user.uid;
41   currentUser.displayName = user.displayName;
42   currentUser.email = user.email;
43   currentUser.photoURL = user.photoURL;
44   return currentUser;
45 }
46 }
```

Cuando el estado de usuario cambie, actualizar observable.

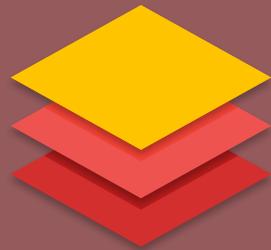
Mapeo objeto a nuestro User.

ACTUALIZANDO HOME COMPONENT

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
export class HomeComponent implements OnInit, OnDestroy {  
  
  currentUser: User;  
  userSub: Subscription;  
  userChecked: boolean;  
  
  constructor(private authService: AuthService) { }  
  
  ngOnInit() {  
    this.userSub = this.authService.currentUser$.subscribe(user => {  
      this.currentUser = user;  
      this.userChecked = true;  
    });  
  }  
  
  ngOnDestroy() {  
    this.userSub.unsubscribe();  
  }  
  
  login() {  
    this.authService.loginByGoogle();  
  }  
  
  logout() {  
    this.authService.logout();  
    this.currentUser = null;  
  }  
}
```

```
<mat-toolbar color="primary" class="mat-elevation-z6">  
  <span>Angular & Firebase</span>  
  <div class="spacer"></div>  
  
  <div *ngIf="userChecked">  
    <button *ngIf="!currentUser; else userBtn" mat-raised-button color="accent" (click)="login()">Log In</button>  
    <ng-template #userBtn>  
      <button mat-raised-button color="accent">{{currentUser.email}}</button>  
      <button mat-icon-button [matMenuTriggerFor]="#menu">  
        <mat-icon>more_vert</mat-icon>  
      </button>  
      <mat-menu #menu="matMenu">  
        <button mat-menu-item (click)="logout()">Log Out</button>  
      </mat-menu>  
    </ng-template>  
  </div>  
</mat-toolbar>
```



MENSAJES

MESSAGE INTERFACE

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

TS message.ts ✘

```
1 import * as firebase from 'firebase';
2
3 export interface Message {
4     userId: string;
5     text: string;
6     imageURL: string;
7     createdAt: firebase.firestore.FieldValue;
8 }
```

OBTENER MENSAJES

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

home.component.ts 

```
38  private getMessages() {
39    this.db.collection<Message>('chat', ref => ref.orderBy('createdAt')).valueChanges().subscribe((messages) => {
40      this.messages = messages;
41    });
42 }
```

BUBBLE CHAT COMPONENT

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)[bubble-chat.component.ts](#)

```
1 import { Component, Input, OnChanges, SimpleChanges } from '@angular/core';
2 import { Message } from '../common/message';
3 import { User } from '../common/user';
4
5 @Component({
6   selector: 'app-bubble-chat',
7   templateUrl: './bubble-chat.component.html',
8   styleUrls: ['./bubble-chat.component.scss']
9 })
10 export class BubbleChatComponent implements OnChanges {
11
12   @Input() currentUser: User;
13   @Input() message: Message;
14   @Input() isMine: boolean;
15
16   defaultAvatar = 'https://ih1.redbubble.net/image.423215377.9237/flat,1200
17
18   ngOnChanges(changes: SimpleChanges): void {
19     if (changes.message && changes.message.currentValue) {
20       if (!this.message.imageURL) {
21         this.message.imageURL = this.defaultAvatar;
22       }
23     }
24   }
25 }
```

[bubble-chat.component.html](#)

```
1 <div class="bubble-chat-wrapper" [ngClass]="{'mine': isMine}">
2   <span class="avatar" [style.backgroundImage]="'url('+ message.imageURL +')'"></span>
3   <p class="message mat-elevation-z6">{{ message.text }}</p>
4   <span class="date">{{ message.createdAt.toDate() | date:'HH:mm' }}</span>
5 </div>
```

BUBBLE CHAT COMPONENT + THEMING

CREANDO PROYECTO

LOGIN BY GOOGLE

MENSAJES

NOTIFICACIONES

DEPLOYMENT

my-theme.scss x

```
1  @import '~@angular/material/theming';
2  @include mat-core();
3
4  $primary: mat-palette($mat-light-blue);
5  $accent: mat-palette($mat-amber);
6  $warn: mat-palette($mat-red);
7  $theme: mat-light-theme($primary, $accent, $warn);
8  $bg: map-get($theme, background);
9  $fg: map-get($theme, foreground);
10
11 @import './app/bubble-chat/bubble-chat.component.sccs-theme.scss';
12
13 @include angular-material-theme($theme);
```

bubble-chat.component.sccs-theme.scss x

```
1  .bubble-chat-wrapper {
2    .message {
3      background-color: mat-color($bg, card);
4      &::before {
5        border-color: transparent mat-color($bg, card) transparent transparent;
6      }
7    }
8    .date {
9      color: mat-color($fg, hint-text);
10   }
11  // customization for mine
12  &.mine {
13    .message {
14      background-color: mat-color($primary);
15      color: mat-color($primary, default-contrast);
16      &::before {
17        border-color: mat-color($primary) transparent transparent transparent;
18      }
19    }
20  }
21 }
```

ACTUALIZANDO HOME COMPONENT

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)[home.component.ts](#)

```
27  ngOnInit() {  
28    this.userSub = this.authService.currentUser$.subscribe  
29    this.currentUser = user;  
30    this.userChecked = true;  
31    if (this.currentUser) {  
32      this.getMessages();  
33    }  
34  );  
35}  
36  
37  ngOnDestroy() {  
38    this.userSub.unsubscribe();  
39  }  
40  
41  private getMessages() {  
42    this.db.collection<Message>('chat', ref => ref.orderBy  
43      this.messages = messages;  
44    }  
45  
46  }  
47  
48  isMine(message: Message) {  
49    return message.userId === this.currentUser.uid;  
50  }  
51  
52  login() {
```

[home.component.html](#)

```
1   <mat-toolbar color="primary" class="mat-elevation-z6">  
2     <span>Angular & Firebase</span>  
3     <div class="spacer"></div>  
4  
5     <div *ngIf="userChecked">...  
6     </div>  
7   </mat-toolbar>  
8  
9   <div class="messaging-area" *ngFor="let message of messages">  
10     <app-bubble-chat [currentUser]="currentUser" [message]="message" [isMine]="isMine(message)">  
11     </app-bubble-chat>  
12   </div>
```

LIMITAR ACCESO

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
export class HomeComponent implements OnInit, OnDestroy {  
  
  currentUser: User;  
  userSub: Subscription;  
  userChecked: boolean;  
  
  messages: Message[];  
  messagesReady: boolean;  
  
  constructor(  
    private db: AngularFirestore,  
    private authService: AuthService) { }  
  
  ngOnInit() { ...  
  }  
  
  ngOnDestroy() { ...  
  }  
  
  private getMessages() {  
    this.db.collection<Message>('chat', ref => ref.orderBy  
      this.messages = messages;  
      this.messagesReady = true; ←  
    });  
  }  
}
```

```
home.component.html ×  
1  <mat-toolbar color="primary" class="mat-elevation-z6">...  
17 </mat-toolbar>  
18  
19 <div *ngIf="userChecked"  
20   <div class="messaging-area" *ngIf="currentUser; else notAcess">  
21     <mat-progress-bar mode="indeterminate" *ngIf="!messagesReady"></mat-progress-bar>  
22     <div class="messages" *ngFor="let message of messages">  
23       <app-bubble-chat [currentUser]="currentUser" [message]="message" [isMine]="isMine(message)">  
24         </app-bubble-chat>  
25     </div>  
26   </div>  
27  
28   <ng-template #notAcess>  
29     <app-not-access></app-not-access>  
30   </ng-template>  
31 </div>
```

ENVIAR MENSAJES

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)[send-message.component.ts](#)

```
1 import { Component, Output, EventEmitter } from '@angular/core';
2
3 @Component({
4   selector: 'app-send-message',
5   templateUrl: './send-message.component.html',
6   styleUrls: ['./send-message.component.scss']
7 })
8 export class SendMessageComponent {
9
10   newMessage: string;
11   @Output() newMessageEvt = new EventEmitter<string>();
12
13   sendMessage(text: string) {
14     if (text) {
15       this.newMessageEvt.emit(text);
16       this.reset();
17     }
18   }
19
20   private reset() {
21     this.newMessage = '';
22   }
23 }
```

[send-message.component.html](#)

```
1 <mat-form-field class="new-message-area">
2   <div>
3     <textarea matInput [(ngModel)]="newMessage" placeholder="Send message"></textarea>
4   </div>
5 </mat-form-field>
6 <div class="sender">
7   <button mat-fab (click)="sendMessage(newMessage)">
8     <mat-icon>send</mat-icon>
9   </button>
10 </div>
```

ACTUALIZANDO HOME COMPONENT

CREANDO PROYECTO

LOGIN BY GOOGLE

MENSAJES

NOTIFICACIONES

DEPLOYMENT

home.component.ts x

```
49  createMessage(text: string) {
50    this.db.collection('chat').add(<Message>{
51      text,
52      imageURL: this.currentUser.photoURL,
53      userId: this.currentUser.uid,
54      createdAt: firebase.firestore.FieldValue.serverTimestamp()
55    });
56 }
```

home.component.html x

```
1  <mat-toolbar color="primary" class="mat-elevation-z6">...
17 </mat-toolbar>
18
19 <div *ngIf="userChecked">
20   <div class="messaging-area" *ngIf="currentUser; else notAcess">
21     <mat-progress-bar mode="indeterminate" *ngIf="!messagesReady"></mat-progress-bar>
22     <div class="messages" *ngFor="let message of messages">
23       <app-bubble-chat [currentUser]="currentUser" [message]="message" [isMine]="isMine">
24     </div>
25     <app-send-message (newMessageEvt)="createMessage($event)"></app-send-message>
26   </div>
27
28 <ng-template #notAcess>
```



NOTIFICACIONES

AGREGANDO MANIFEST Y FIREBASE SW

CREANDO PROYECTO

LOGIN BY GOOGLE

MENSAJES

NOTIFICACIONES

DEPLOYMENT

{} manifest.json x

```
1 {  
2   "short_name": "AngularFirebaseGDG",  
3   "name": "Angular Firebase GDG",  
4   "start_url": "/?utm_source=homescreen",  
5   // SIEMPRE MISMO ID  
6   "gcm_sender_id": "103953800507"  
7 }
```

JS firebase-messaging-sw.js x

```
1 importScripts('https://www.gstatic.com/firebasejs/3.9.0.firebaseio-app.js');  
2 importScripts('https://www.gstatic.com/firebasejs/3.9.0/firebase-messaging.js');  
3  
4  
5 firebase.initializeApp({  
6   messagingSenderId: '26732418254'  
7 });  
8  
9 const messaging = firebase.messaging();
```

TS environment.ts x

```
1 export const environment = {  
2   production: false,  
3   firebase: {  
4     apiKey: 'AIzaSyAPil2za_u2I-a8aln44xFv33djxSLVss',  
5     authDomain: 'angularfirebasergd.firebaseio.com',  
6     databaseURL: 'https://angularfirebasergd.firebaseio.com',  
7     projectId: 'angularfirebasergd',  
8     storageBucket: 'angularfirebasergd.appspot.com',  
9     messagingSenderId: '26732418254'  
10   }  
11 };
```

A angular.json x

```
21   "index": "src/index.html",  
22   "main": "src/main.ts",  
23   "polyfills": "src/polyfills.ts",  
24   "tsConfig": "src/tsconfig.app.json",  
25   "assets": [  
26     "src/favicon.ico",  
27     "src/assets",  
28     "src/firebase-messaging-sw.js",  
29     "src/manifest.json"  
30   ],  
31   "styles": [  
32     "src/styles.scss"  
33   ].
```

MESSAGING SERVICE

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
A messaging.service.ts x
  8  export class MessagingService {
  9
 10    private messaging = firebase.messaging();
 11    private currentToken: string;
 12
 13    constructor(private http: HttpClient) { }
 14
 15    getPermission() {
 16      this.messaging.requestPermission()
 17        .then(() => {
 18          console.log('Permisos de Notificación aceptado.');
 19          this.messaging.getToken().then((token) => {
 20            console.log(token);
 21            this.currentToken = token;
 22          });
 23        });
 24    }
 25
 26    sendNotification() {
 27      const request = {
 28        notification: {
 29          title: 'Hola Notificaciones!',
 30          body: 'Esto es una notificación desde Firebase usando Angular',
 31          icon: '../../../../../assets/logo.png',
 32          click_action: 'https://alejandrolora.com',
 33        },
 34        to: this.currentToken
 35      };
 36      return this.http.post('https://fcm.googleapis.com/fcm/send', request,
 37      {
 38        headers: {
 39          'Content-Type': 'application/json; charset=utf-8',
 40          // messaging- SERVER-key
 41          'Authorization': 'key=AAAABjlgVM4:APA91bFmk556D3KzSmP-Biwm2a0I99c'
 42        }
 43      });
 44    }
 45  }
```

Pide permiso para notificaciones
y guarda el token.

Objeto notificación.
Más abajo agregamos el token.

Usamos el messaging server key.

MESSAGING SERVICE

CREANDO PROYECTO

LOGIN BY GOOGLE

MENSAJES

NOTIFICACIONES

DEPLOYMENT

Messaging Server Key

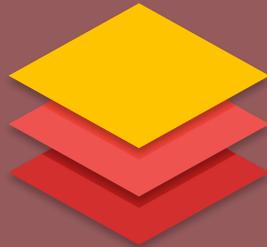
The screenshot shows the Firebase console interface for a project named "my-app". A red arrow points from the top-left towards the "Configuración del proyecto" button in the navigation bar. Another red arrow points from the bottom-left towards the "Mensajería en la nube" tab in the main navigation bar. A third red arrow points from the bottom-left towards the "Clave de servidor" row in the "Credenciales de proyecto" table, which contains the server key value.

Clave	Token
Clave de servidor	AAAAAHNg64f4:APA91bGysfJn65XTmgHOEM2JMwgwLz9flKP9MT8gExVPDSRFy_XLrRwCAb0lxqbtPZmvuNa5Flz84JBSdT691k5h2flunKx5T9i3swiHRWixJ8WhPdYy6qxfMSDVdj9CV3nlbtmqbQUAVUkdbgSLaY-uU-Elk6b_w
Clave de servidor heredada	AlzaSyAezxsEiRITUJ-gTajs8L96RecAw2BdE-I
ID del remitente	123928764926

ACTUALIZANDO HOME COMPONENT

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
A home.component.ts x
  25 |   constructor(
  26 |     private db: AngularFirestore,
  27 |     private authService: AuthService,
  28 |     private messagingService: MessagingService) { }
  29 |
  30 |   ngOnInit() {
  31 |     this.userSub = this.authService.currentUser$.subscribe(user => {
  32 |       this.currentUser = user;
  33 |       this.userChecked = true;
  34 |       if (this.currentUser) {
  35 |         this.getMessages();
  36 |         this.messagingService.getPermission();
  37 |       }
  38 |     });
  39   }
  40 |
  41 |   ngOnDestroy() { ... }
  42 |
  43 |   private getMessages() { ... }
  44 |
  45 |   createMessage(text: string) { ... }
  46 |
  47 |   sendNotification() {
  48 |     setTimeout(() => this.messagingService.sendNotification().subscribe(), 4000);
  49 |   }
  50 |
  51 | }
```



DEPLOYMENT (& MEJORAS)

ENTER EVENT

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
5 send-message.component.html ×
1   <mat-form-field class="new-message-area">
2     <div>
3       <textarea matInput [(ngModel)]="newMessage" (keyup.enter)="sendMessage(newMessage)" placeholder="Send message">
4     </textarea>
5   </div>
6 </mat-form-field>
7 <div class="sender">
8   <button mat-fab (click)="sendMessage(newMessage)">
9     <mat-icon>send</mat-icon>
10    </button>
11 </div>
```

ACTUALIZANDO MESSAGE

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
TS message.ts ×  
1 import { firestore } from 'firebase/app';  
2  
3 export interface FirebaseTime extends firestore.FieldValue {  
4   toDate(): Date;  
5 }  
6  
7 export interface Message {  
8   userId: string;  
9   text: string;  
10  imageURL: string;  
11  createdAt: FirebaseTime;  
12 }
```

```
5 bubble-chat.component.html ×  
1 <div class="bubble-chat-wrapper" [ngClass]="{'mine': isMine}">  
2   <span class="avatar" [style.backgroundImage]="'url('+ message.imageURL +')'"></span>  
3   <p class="message mat-elevation-z6">{{ message.text }}</p>  
4   <span class="date">{{ message.createdAt?.toDate() | date:'HH:mm' }}</span>  
5 </div>
```

ACTUALIZANDO HOME COMPONENT

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

```
home.component.ts ×
26  @ViewChild('messagingArea') messagingArea: ElementRef;
27  @ViewChildren('chats') chats: QueryList<any>;
28
29  constructor(
30    private db: AngularFirestore,
31    private authService: AuthService,
32    private messagingService: MessagingService) { }
33
34  ngOnInit() { ... }
35
36  ngAfterViewInit() {
37    this.messagesSub = this.chats.changes.subscribe(() => this.scrollToBottom());
38  }
39
40  ngOnDestroy() {
41    this.userSub.unsubscribe();
42    this.messagesSub.unsubscribe();
43  }
44
45  private scrollToBottom() {
46    this.messagingArea.nativeElement.scrollTop = this.messagingArea.nativeElement.scrollHeight
47  }
48
49
50
51
52
53
54
55
56
```

```
home.component.html ×
17  <div *ngIf="userChecked">
18    <div class="messaging-area" #messagingArea *ngIf="currentUser; else notAccess">
19      <mat-progress-bar mode="indeterminate" *ngIf="!messagesReady"></mat-progress-bar>
20      <div class="messages" *ngFor="let message of messages">
21        <app-bubble-chat #chats [currentUser]="currentUser" [message]="message" [isMine]="'>
22      </div>
23      <app-send-message (newMessageEvt)="createMessage($event)"></app-send-message>
24    </div>
```

PERSISTENCIA LOGIN BY GOOGLE

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

A auth.service.ts ×

```
18  loginByGoogle() {
19    const provider = new auth.GoogleAuthProvider();
20    this.afAuth.auth.setPersistence(auth.Auth.Persistence.SESSION).then(() => {
21      this.afAuth.auth.signInWithPopup(provider);
22    });
23  }
```

FIJANDO VERSIONES

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)

package.json x

```
13 "dependencies": {  
14   "@angular/animations": "6.0.6",  
15   "@angular/cdk": "6.3.0",  
16   "@angular/common": "6.0.6",  
17   "@angular/compiler": "6.0.6",  
18   "@angular/core": "6.0.6",  
19   "@angular/forms": "6.0.6",  
20   "@angular/http": "6.0.6",  
21   "@angular/material": "6.3.0",  
22   "@angular/platform-browser": "6.0.6",  
23   "@angular/platform-browser-dynamic": "6.0.6",  
24   "@angular/router": "6.0.6",  
25   "angularfire2": "5.0.0-rc.11",  
26   "core-js": "2.5.7",  
27   "firebase": "5.1.0",  
28   "hammerjs": "2.0.8",  
29   "rxjs": "6.2.1",  
30   "zone.js": "0.8.26"  
31 },
```

Nunca vayas a producción sin fijar las versiones de las dependencias.

Caret ^ : 6.X.X

Tilde ~ : 6.0.X

FIREBASE CLI

[CREANDO PROYECTO](#)[LOGIN BY GOOGLE](#)[MENSAJES](#)[NOTIFICACIONES](#)[DEPLOYMENT](#)[PROBLEMS](#) [OUTPUT](#) [DEBUG CONSOLE](#) [TERMINAL](#)

```
mbpdealejandro:angular-firebase-gdg alejandrolora$ npm install -g firebase-tools
/usr/local/bin/firebase => /usr/local/lib/node_modules/firebase-tools/bin/firebase
+ firebase-tools@3.19.1
updated 1 package in 4.089s
mbpdealejandro:angular-firebase-gdg alejandrolora$ firebase init
```



You're about to initialize a Firebase project in this directory:

```
/Users/alejandrolora/Desktop/angular-firebase-gdg
```

```
? Which Firebase CLI features do you want to setup for this folder? Press Space to select features,
then Enter to confirm your choices. Hosting: Configure and deploy Firebase Hosting sites
```

```
== Project Setup
```

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running `firebase use --add`,
but for now we'll just set up a default project.

```
? Select a default Firebase project for this directory: AngularFirebaseGDG \(angularfirebasegdg\)
```

```
== Hosting Setup
```

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with `firebase deploy`. If you
have a build process for your assets, use your build's output directory.

```
? What do you want to use as your public directory? dist/angular-firebase-gdg
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
✓ Wrote dist/angular-firebase-gdg/index.html
```

```
i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...
```

```
✓ Firebase initialization complete!
```

```
mbpdealejandro:angular-firebase-gdg alejandrolora$ [ ]
```

```
npm install -g firebase-tools
```

```
firebase init
```

DEPLOYMENT

CREANDO PROYECTO

LOGIN BY GOOGLE

MENSAJES

NOTIFICACIONES

DEPLOYMENT

```
🔥 .firebaserc ✘
1  {
2    "projects": {
3      "default": "angularfirebasegdg"
4    }
5 }
```

```
{ } firebase.json ✘
1  {
2    "hosting": {
3      "public": "dist/angular-firebase-gdg",
4      "ignore": [
5        "firebase.json",
6        "**/*.",
7        "**/node_modules/**"
8      ],
9      "rewrites": [
10        {
11          "source": "**",
12          "destination": "/index.html"
13        }
14      ]
15    }
16 }
```

```
package.json ✘
1  {
2    "scripts": {
3      "ng": "ng",
4      "start": "ng serve",
5      "build": "ng build",
6      "test": "ng test",
7      "lint": "ng lint",
8      "e2e": "ng e2e",
9      "deploy": "ng build --c=production && firebase deploy && firebase open"
10    },
11    "postinstall": "ng build --prod"
12  },
13  "dependencies": {
14    "angular": "5.2.10",
15    "angularfire2": "5.0.0-beta.11",
16    "firebase": "5.0.1",
17    "rxjs": "5.5.2",
18    "core-js": "2.5.4"
19  },
20  "devDependencies": {
21    "angular-cli": "1.7.4",
22    "codelyzer": "3.2.1",
23    "jasmine-core": "3.1.0",
24    "jasmine-spec-reporter": "3.4.1",
25    "karma": "3.0.0",
26    "karma-chrome-launcher": "2.2.2",
27    "karma-jasmine": "1.1.1",
28    "karma-jasmine-html-reporter": "0.2.2",
29    "protractor": "5.4.1",
30    "tslint": "5.9.1",
31    "typescript": "2.9.2"
32  }
33 }
```



LIVE DEMO

<https://angularfirebaseqdg.firebaseioapp.com/>

<https://github.com/AL-MeetUps/AngularFirebaseGDG>



¡GRACIAS!

¿PREGUNTAS?