Chat con Ionic 3 + Firebase

Sara Zapico Fernández

Ionic 3

- Framework de desarrollo de aplicaciones móviles utilizando tecnologías web
  - Angular
  - Typescript
  - SASS
  - HTML 5
- Plataformas soportadas
  - Android
  - IOS
- Links
  - Documentación: https://ionicframework.com/
  - Consola de administración: https://dashboard.ionicjs.com/apps
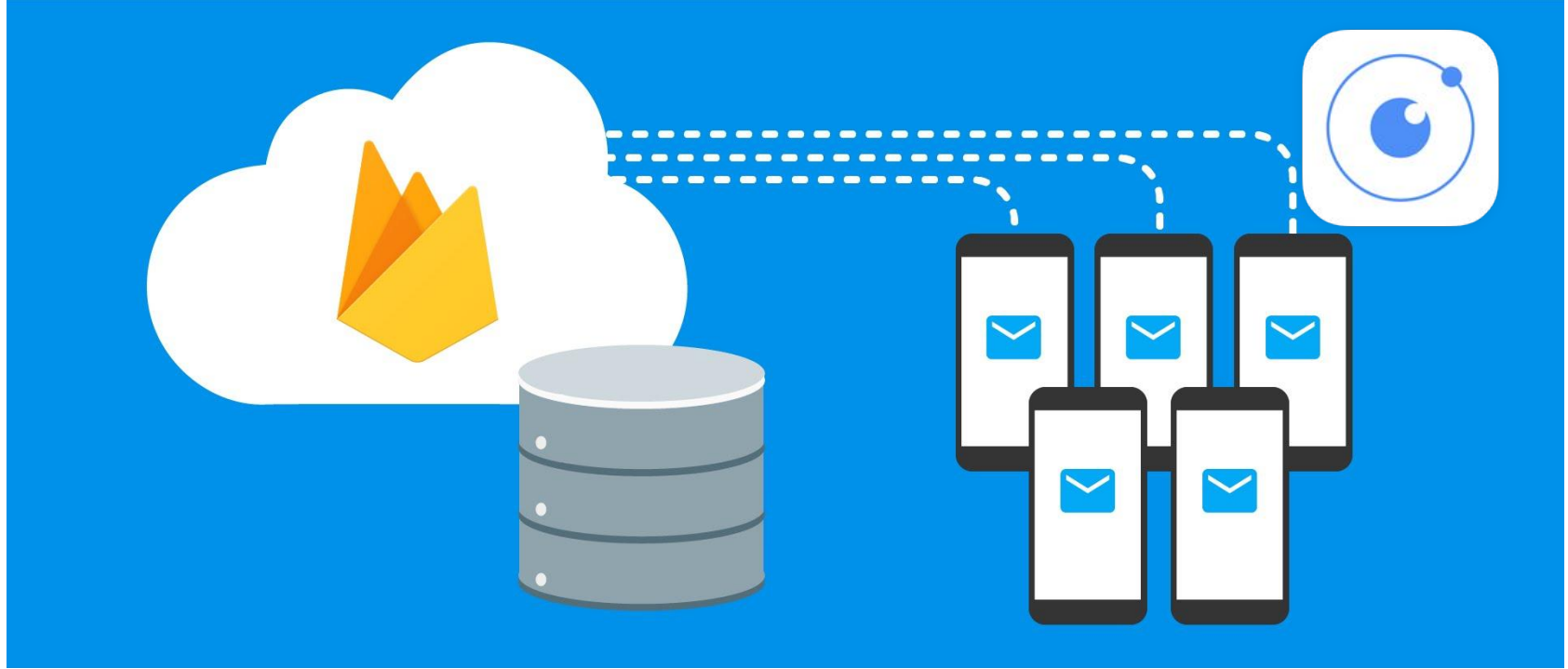
Firebase

- Plataforma, de Google, para el desarrollo de aplicaciones web y aplicaciones móviles
- Servicios como:
  - Firebase Cloud Messaging (notificaciones)
  - **Firebase Auth (autenticación)**
  - **Realtime Database y Firebase Firestore (base de datos)**
  - Firebase Storage (almacenamiento de ficheros)
- Link consola administración
  - https://console.firebase.google.com

Chat = Ionic 3 + Firebase

Primeros pasos

- Clonar el proyecto: git clone https://github.com/sarazafe/firebase-ionic-chat.git
- Ir a la carpeta del proyecto e instalar el plugin de firebase
  - npm install --save firebase
  - Añadir al proyecto → **import * as Firebase from 'firebase';** en **app.components.ts**
  - Inicializar firebase → **firebase.initializeApp(FirebaseConfig);** en el constructor de **app.components.ts**
  - Añadir la configuración de Firebase en **firebase-config.ts**

```
export const FirebaseConfig = {
    apiKey: "AIzaSyAK2aBe_WODidZW9lCqIFQPI5eXOTpFvNU",
    authDomain: "chat-59245.firebaseapp.com",
    databaseURL: "https://chat-59245.firebaseio.com",
    projectId: "chat-59245",
    storageBucket: "chat-59245.appspot.com",
    messagingSenderId: "561348409962"
};
```

Darse de alta

- LoginProvider
  - signUp

```
return firebase.auth().createUserWithEmailAndPassword(user.email,
user.password);
```

- LoginPage
  - signUp

```
this.userProvider.saveUser(user).then(() => {
        this.navCtrl.push(ChatPage, {
         user: user
});
```

Login

- LoginProvider
  - login

```
return this.auth.signInWithEmailAndPassword(user.email,user.password);
```

- LoginPage
  - doLogin

```
user.uid = value.uid;
this.navCtrl.push(ChatPage, {
    user: user
});
```

Iniciar room

- RoomProvider
  - initRoom

    ```
    return this.database.ref(Endpoints.ROOMS + Chat.DEFAULT_ROOM_ID).set(new Room(Chat.DEFAULT_ROOM_ID, Chat.DEFAULT_ROOM_NAME));
    ```

Iniciar room (II)

- ChatPage
  - initRoom

```
this.roomProvider.initRoom().then(() => {
    // Add user to the room it he/she has not been added
    if (!this.member) {
     this.member = new Member({
      uid: this.user.uid,
      email: this.user.email,
      roomId: Chat.DEFAULT_ROOM_ID
     });      this.memberProvider.addMember(this.member)
      .then().catch(function (error) {
      console.log("Error adding member", error);
     });
    }
}).catch(function (error) {
    console.log("Init room error", error);
   });
```

Añadir usuario al chat

- MemberProvider
  - addMember

```
return this.database.ref(Endpoints.MEMBERS + member.roomId + "/" + member.uid).set(member);
```

- ChatPage
  - onNewMember

```
this.memberProvider.getMemberReference().on('child_added', (val) => {
        let member: Member = new Member(val.val());
        if (this.user.email === member.email) {
                this.member = member;
        }
        this.members.push(new Member(val.val()));
});
```

Enviar mensajes

- MessageProvider
  - saveMessage

```
return this.getMessageReference().push().set(message);
```

- ChatPage
  - sendMessage

```
this.messageProvider.saveMessage(message)
 .then(()=> {
   // Clean sent message
   this.message = '';
 }).catch(function (error) {
 console.log("Error saving message", error);
 });
```

Recibir mensajes

- ChatPage
  - onNewMessage

```
this.messageProvider.getMessageReference().on('child_added', (val) => {
 let message: Message = new Message(val.val());
 if (this.member.email === message.sender) {
  message.position = 'right';
 } else {
  message.position = 'left';
 }
 this.messages.push(message);
});
```