

VIRAVI

Autores:

Víctor Henry Fall Turrientes

Raúl Arribas Canalejo

Víctor Corchado Guirao

Tutor:

Diego Baratto Valdivia

CFGS Desarrollo de Aplicaciones Multiplataforma

Curso 2022-2023

Innovación en Formación Profesional

Convocatoria de Presentación: mayo 2023

Índice

1.	Introducción	4
2.	Justificación del proyecto.....	5
3.	Objetivo general y objetivos específicos	5
4.	Estado del arte	6
5.	Métodos de trabajo.....	7
6.	Resultado	8
7.	Conclusiones	21
8.	Bibliografía.	22

VIRAVI es una aplicación innovadora, original, distintiva y con una amplia gama de características y funcionalidades. Esta aplicación ha sido desarrollada con el objetivo de crear una red social en la que la principal característica es la creación de eventos de todo tipo para hacer el mundo más sociable y conectar a miles de personas. Su creación ha sido fruto del trabajo, esfuerzo y sobre todo constancia, aunque es cierto que aún tiene mucho que progresar y mejorar para llegar a ser la mejor red social de eventos. Dentro de VIRAVI se pueden encontrar todo tipo de actividades, tales como: eventos deportivos, socioculturales, gastronómicos, informativos e incluso turísticos. Esta aplicación se ha desarrollado por tres estudiantes del CFGS de DAM utilizando Android Studio como IDE y Java como lenguaje de programación, poniendo en práctica, de esta manera, lo aprendido durante el ciclo formativo. Además, de forma externa, se ha empleado Firebase como base de datos y GitHub como administrador de versiones. Se espera que VIRAVI llegue a completarse de forma definitiva en un futuro no muy lejano y sea posible su introducción en el mercado, aunque, bien es cierto que, para lograr esta meta, aún se necesita mucho conocimiento, esfuerzo y dedicación.

Con todo lo anterior y sin más preámbulo, se presenta VIRAVI, la red social creada para conocer gente con tus mismos intereses y vivir experiencias únicas e inolvidables.

1. Introducción

VIRAVI es una aplicación para dispositivos móviles categorizada como red social, en la cual se podrán encontrar eventos y actividades de todo tipo organizados por los propios usuarios pertenecientes, donde podrá apuntarse cualquier persona que disponga de un dispositivo móvil y de muchas ganas de divertirse.

Este proyecto tiene como objetivo que nadie quede excluido y que, sea cual sea la preferencia del cliente, siempre haya un evento que pueda satisfacer sus necesidades.

Dentro de VIRAVI se podrá encontrar eventos de las siguientes categorías:

- **Eventos deportivos** donde no sólo se busca la diversión, sino también se busca fomentar la buena salud y la pasión por el deporte. Las opciones más destacadas y populares suelen ser:
 - Exhibiciones de arte urbano, paracaidismo, carreras solidarias, ciclismo, escalada, piragüismo, etc.
- **Eventos gastronómicos.** Se debe promocionar y gozar de la maravillosa y polivalente dieta mediterránea y, fusionándola con alta cocina moderna, se sabe con seguridad que los usuarios vivirán una experiencia nunca antes vivida donde los 5 sentidos estarán completamente abrumados y sobrepasados con una cantidad inigualable de sensaciones. En cuanto a los eventos más comunes entre la gente con buen paladar, destacan:
 - Alta cocina mediterránea, Mercadillos gastronómicos, catas, experiencias gourmet, degustaciones, etc.
- **Eventos culturales.** Una amplia gama de experiencias que de seguro causarán muchas emociones a los clientes.
 - Galerías de arte, museos, espectáculos, exhibiciones al aire libre y muchas otras actividades para todo tipo de público.
- **Eventos musicales.** Se ha seleccionado cuidadosamente cada uno de los eventos y sus diversos géneros. Teniendo esto en cuenta, los clientes podrán disfrutar de ellos como nunca. Los más populares son:
 - Conciertos, festivales, musicales, karaokes, etc.
- **Eventos informativos.** Para la búsqueda de información y del enriquecimiento personal, los clientes tendrán acceso a una gran cantidad de reuniones, convenciones, debates, investigaciones... Acerca de todos los temas más interesantes, novedosos e inquietantes.

2. Justificación del proyecto

La principal motivación que permitió llevar a cabo este proyecto fue la necesidad común de hacer planes con grandes cantidades de personas, ésta encontrada en distintos grupos sociales, cada uno con unas particularidades diferentes, y que tras llevar a cabo un pequeño análisis de situación desencadenaron en VIRAVI.

Además de apoyar bienes no lucrativos, esta aplicación fomenta el bienestar tanto personal como grupal. Por ende, se creó VIRAVI, con unas posibilidades infinitas de creación de eventos por el territorio español que permiten a las personas conocerse e integrarse en la sociedad de una forma divertida en función de sus necesidades específicas.

Con relación a lo anterior, concienciarse de que la vida a veces puede ser estresante, o simplemente la necesidad de desconectar de todas las responsabilidades para tener momentos de confraternidad, son motivos más que suficientes para que esta red social se pusiese en funcionamiento.

3. Objetivo general y objetivos específicos

El principal objetivo de VIRAVI es ser una plataforma colectiva la cual permita en un futuro a corto plazo, ayudar a personas a encontrar la felicidad y hacer del mundo un lugar más sociable.

Tras ello hay un gran esfuerzo que ha permitido que esta red social se haya consumado. Para poder llegar al lugar en el que está actualmente se han estudiado tecnologías, desarrollado aptitudes con las que no se contaban al inicio y sobre todo se ha puesto mucho esfuerzo y sacrificio.

Algunos de los objetivos específicos que se propusieron al comienzo del desarrollo fueron:

- Desarrollar una base de datos no relacional en Firebase.
- Realizar importación y exportación de datos a Firebase a través de la aplicación.
- Realizar un registro de usuarios y eventos a la base de datos de Firebase con sus pertinentes validaciones.
- Conseguir un inicio de sesión con la autenticación de Firebase para lograr mantener la sesión iniciada y cerrar sesión en cualquier momento.
- Aprender a almacenar imágenes en Firebase para su posterior uso dentro de la aplicación.

- Registrar eventos de cualquier categoría como usuario.
- Facilitar un filtro de búsqueda de eventos mediante la categoría particular de cada uno de ellos.
- Conseguir una interfaz sencilla con distintos apartados con la cual el usuario pueda interactuar de manera intuitiva.
- Crear un apartado para el perfil del usuario y su información personal.
- Mostrar en la parte del perfil del usuario, los eventos que el mismo ha creado.
- Registrar usuarios en eventos previamente creados.
- Mostrar en la parte del perfil del usuario, los eventos a los que se ha apuntado.
- Seguir y/o dejar de seguir a distintos usuarios.
- Reflejar los seguidores y los usuarios que sigue cada usuario en el apartado de perfil del usuario.
- Conseguir un chat a tiempo real, que fuese funcional, entre dos usuarios dentro de la red social.
- Crear una interfaz nueva para visualizar los chats de cada usuario con los creadores de los eventos.

4. Estado del arte

Una vez realizado un exhaustivo análisis de mercado que mostró muchas aplicaciones con usos y utilidades parecidas a las que VIRAVI pretendía responder, se decidió que las tres aplicaciones de eventos, posteriormente presentadas, serían las que servirían de ejemplo para llevar a cabo la aplicación de eventos que se sitúa en este contexto:

En primer lugar, se presenta Whova, aplicación ganadora del premio “Best EventApp”. Sabiendo que ésta era la mejor aplicación de eventos actualmente, no se podía dejar pasar la oportunidad de inspirarse en ella.

Su eslogan es, “Make your attendees 10 times happier” (Whova. 2023) y en base a ello, se decidió que los clientes de VIRAVI debían ser diez veces más felices tras el uso y disfrute de la aplicación.

A continuación, el equipo se percató que para ganarse la atención de los clientes sería necesario que la aplicación a desarrollar llamase la atención del público específico al que se enfocaría, presentase un diseño intuitivo y fuese fácil de utilizar. Por todo lo anterior, para la parte del diseño de la aplicación, la distribución de los eventos y la manera en la que el usuario puede navegar por la interfaz, se hizo énfasis en el enfoque que Aloomba da

a su aplicación. Una aplicación sencilla, que vista desde dentro, como se observa en su web (Aloompa. 2023), presenta un diseño minimalista que permite ser muy atractivo e interactivo con el usuario. El diseño de esta aplicación fue muy inspirador para realizar el aspecto externo de VIRAVI, el cual pretende ser organizado, interactivo, funcional y elegante.

Para finalizar, se muestra Meetup (Meetup. 2023), una aplicación centrada en descubrir grupos y eventos locales para conocer personas, hacer distintos planes, divertirse socializando o simplemente encontrar gente para realizar actividades cotidianas como ir al gimnasio o hacer un picnic. El símil que presenta VIRAVI en concordancia con esta aplicación, es la de compartir momentos apasionantes, conocer gente, gustos e intereses similares y disfrutar de maravillosas experiencias. Lo cual se aplicó a VIRAVI de forma prácticamente idéntica porque son los valores que inicialmente presentaba la red social de eventos que se planeó desarrollar.

5. Métodos de trabajo

En este apartado se explicará cómo se ha creado VIRAVI, mostrando su planificación interna y demostrando que es un proyecto que, por supuesto, sería viable para presentarse al mercado y competir con los más grandes del sector.

Esta aplicación, ha sido fruto del trabajo y esfuerzo de todos los desarrolladores pertenecientes a este grupo, pero, para llegar al momento actual, se ha tenido que recorrer un abstruso y arriesgado camino:

En primer lugar, se decidió que el proyecto debía ser una aplicación móvil, esto acarreó que los integrantes del grupo tuviesen que estudiar y ampliar sus conocimientos a un nivel superior a lo aprendido en las aulas acerca de Android Studio y la programación en este arduo entorno de desarrollo.

Una vez se escogió el formato de aplicación y el entorno en el cual se iba a desarrollar, se dio la tesitura de realizar la aplicación con Kotlin o con Java y el tanto el tiempo como otros aprendizajes que se tuvieron que dar para que VIRAVI llegase, hicieron que el lenguaje que se iba a emplear fuese Java, un lenguaje conocido por todos los integrantes del proyecto y que sin duda iba a ser lo más rápido a la hora de realizar los pertinentes códigos de desarrollo.

Posteriormente se procedió con la lluvia de ideas, también conocida como “brainstorming”, esta parte del proyecto fue la más tardada junto con la de realización del proyecto que se mostrará próximamente, esto se debió a la cantidad de ideas que aportaban todos los miembros, tales como: una aplicación con valoraciones y recomendaciones de películas, una aplicación donde compartir a tiempo real momentos y videos graciosos con tus amigos etc.

Tras ello, se decidió “unificar” todas las ideas para formar la red social existente en estos momentos.

Cuando se decidió el modelo de aplicación que se iba a realizar, hubo que estudiarse las funcionalidades de Firebase, una base de datos no relacional, a tiempo real y con mucho potencial que sería la utilizada finalmente en el proyecto.

Antes de comenzar a desarrollar, se dio el problema del control de versiones, que planificación se iba a llevar a cabo, como se iba a compartir el código y que formato se iba a utilizar para no perderse entre tanto cambio. Tras una gran investigación, se llegó a la conclusión que lo que se debía hacer, era aprender cómo funcionaba git y en consecuencia github. Esta fue sin duda la herramienta que permitió llevar a cabo un control de versiones sin pisarse el trabajo.

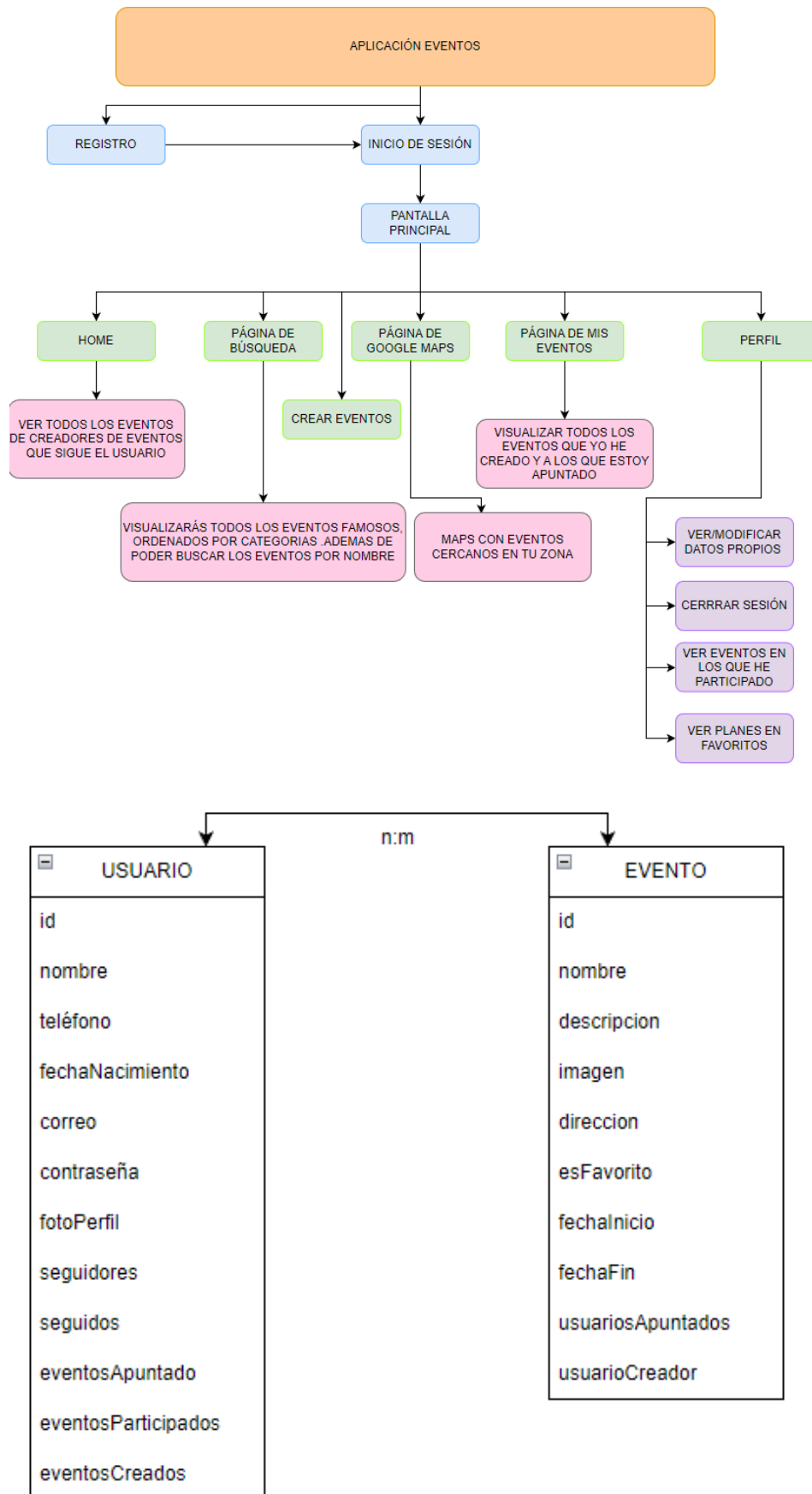
Para finalizar se decidió como se iban a repartir las tareas dentro del proyecto y la parte que le correspondería a cada miembro, estableciéndose unos plazos de entrega internos y unas “reuniones de seguimiento” para comprobar el estado semanal de la aplicación.

6. Resultado

VIRAVI ha pasado por muchas etapas desde que se comenzó a especular sobre ella, pasó de ser una aplicación con una temática un tanto infantil y una interfaz de usuario minimalista, a convertirse en una aplicación con un aspecto más elegante y trabajado. Pero para llegar aquí, primero se debe explicar los comienzos y el cómo surgió VIRAVI.

Las primeras ideas de funcionalidad para esta red social fueron muy simples y sin apenas ambición, se pretendía hacer algo básico y funcional debido a que el tiempo agobiaba y los desarrolladores no se veían capaces de mostrar en la aplicación todas las ideas que iban teniendo dada la complejidad de las mismas, pero poco a poco se empezaron a desarrollar funcionalidades, las ideas fueron haciéndose realidad gracias una buena organización e implicación de los integrantes, y finalmente se llegaron a conseguir cosas que al comenzar el proyecto parecían completamente imposibles para desarrolladores tan inexpertos.

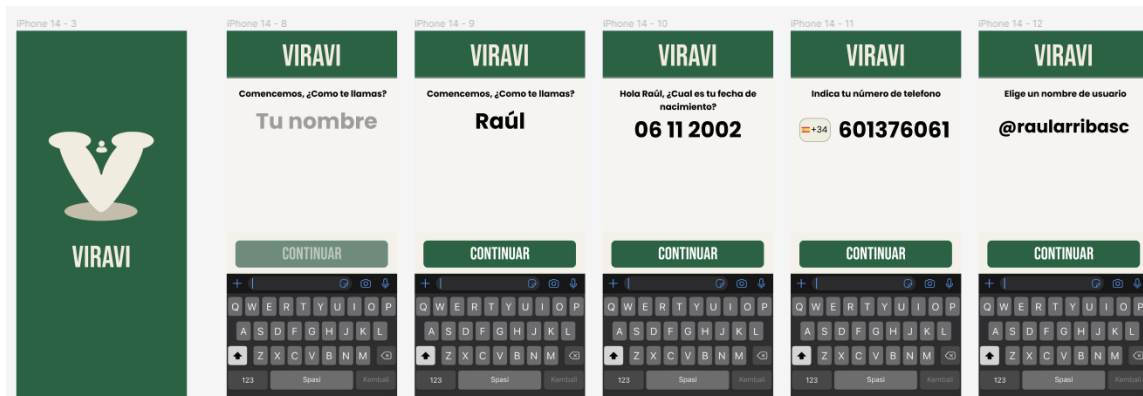
A continuación, se muestra un esquema orientativo con las funcionalidades iniciales que se pretendían presentar en la aplicación y un primer modelo inicial de base de datos que se pensó sin tener en cuenta que Firebase iba a ser la base de datos que se utilizaría en última instancia, la cual es no relacional y por tanto no eran necesarias tablas:



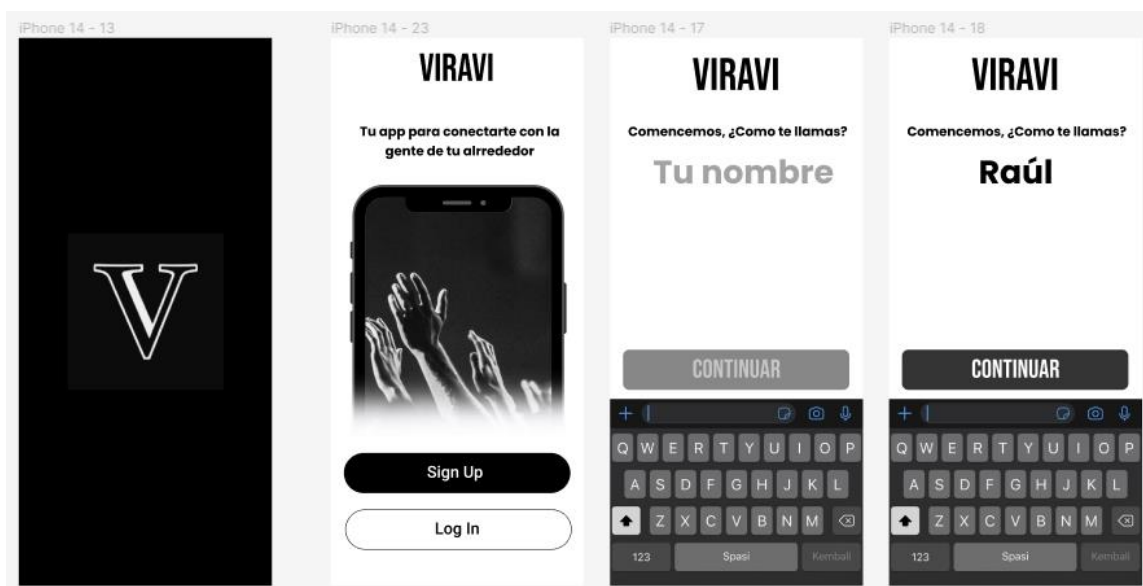
Como puede observarse, el modelo inicial no puede compararse con la aplicación final

entregada. En la imagen anterior, pueden observarse detalles como “Página de Google Maps”, ya que inicialmente se pensó crear un apartado en la interfaz donde hubiese un mapa para conocer la geolocalización de cada uno de los eventos.

Tras todo este modelaje de ideas, se comenzó a diseñar la aplicación en Figma (Figma, s. f.), de esta manera, podría concibirse una idea más acercada a la realidad de como iba a ser finalmente la red social y comenzar a programar las distintas funcionalidades.



En esta imagen puede verse como estaba pensado inicialmente el apartado de registro de usuarios. Pero, tras varias reuniones se plantearon otros diseños más minimalistas, elegantes y para un público más adulto.

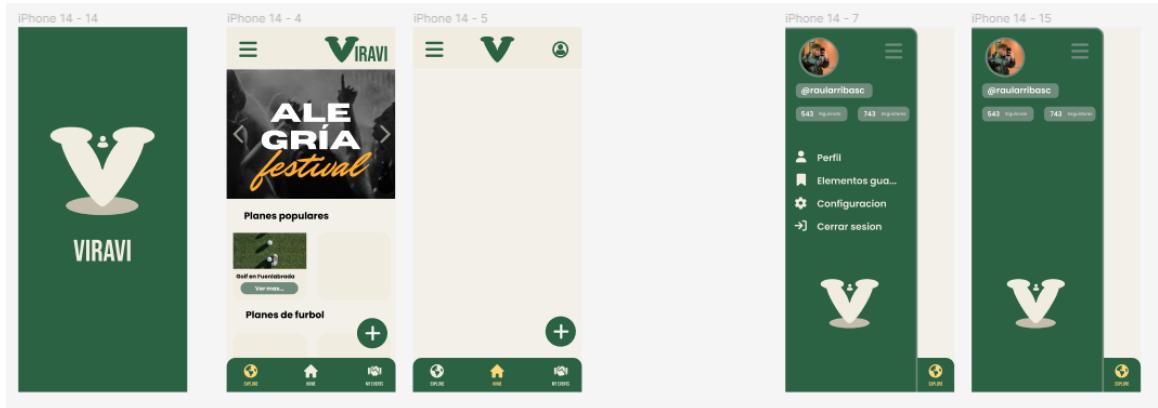


Aun así, este modelo, que ya gustaba más, se varió ligeramente hasta dar con el modelo actual. Recaltar que, los cambios de interfaz, realizados se implementaron prácticamente al final del proceso y que la mayor parte del tiempo de desarrollo de funcionalidades como: registro de usuarios, inicio de sesión, publicación de eventos... fueron desarrollados en el formato original de la aplicación.

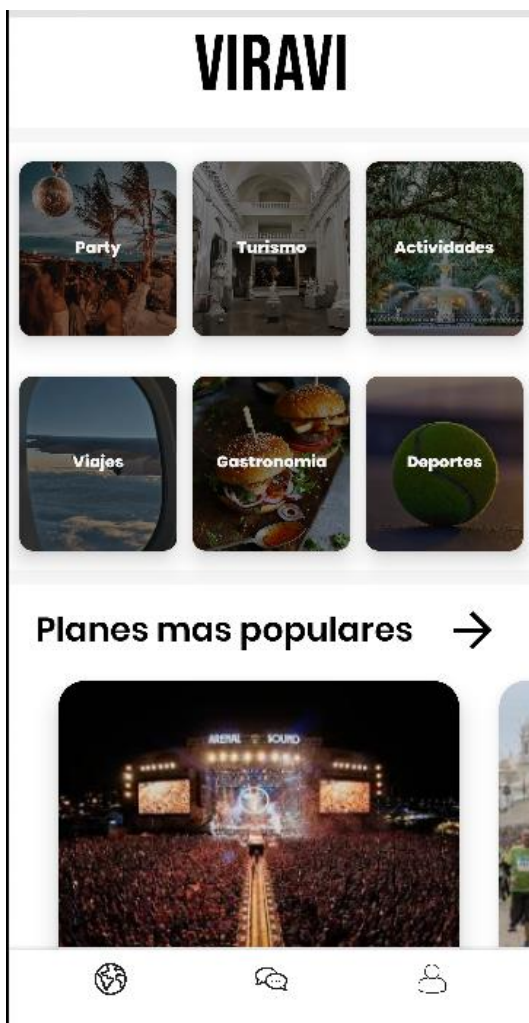
Una vez se pasa el registro de usuarios y el inicio de sesión se muestra una página principal, donde se podrá ver el perfil de los usuarios, ver los eventos más populares e incluso tener un

chat con el creador del evento en cuestión.

La página principal, en un principio, iba a tener un formato similar al mostrado anteriormente con esas tonalidades verdes, diversas fuentes para los textos, mucho color en las imágenes, etc.



Pero tras muchos cambios y arreglos, finalmente quedó con un formato y un diseño mucho más formal, elegante e intuitivo para los usuarios.



En conclusión, para este apartado, el esfuerzo y la dedicación han sido dos de los pilares

fundamentales del grupo, ya que, desde la primera idea, la cual pretendía ser cumplidora, se ha conseguido crear una aplicación perfectamente funcional que, con ciertos detalles a mejorar como la seguridad y actualizaciones de contenido, podría llegar muy lejos en el mercado español y quien sabe si también en el mercado internacional.

Por otro lado, también deben destacarse ciertas partes del proyecto que llevaron mucho tiempo, debido a su complejidad y o desconocimiento de funcionalidad como, por ejemplo: La inserción de datos en Firebase y la posterior extracción de estos para realizar el inicio de sesión o, el visualizado de eventos en la aplicación utilizando fragments. Esto llevó más tiempo de lo necesario debido al poco conocimiento de los programadores sobre Firebase y Android Studio, pero tras un estudio exhaustivo lograron realizarlo. En un principio, fue sencillo realizar el inicio de sesión y el registro de usuario debido a que Firebase tiene sus propios métodos que hacían el trabajo que queríamos. El problema se produjo cuando se insertaron los fragments con el objetivo de agilizar la interacción del usuario en el proceso de registrarse o iniciar sesión, ya que era necesario pasar la información de un fragment a otro para obtener todos los datos recopilados en el proceso y de esta manera almacenarlos en la base de datos.

Inicialmente, el registro de usuarios se realizaba de la siguiente manera:

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    emailAlumno = findViewById(R.id.email);  
    passwordAlumno = findViewById(R.id.password);  
    inicioSesion = findViewById(R.id.inicioSesion);  
  
    firebaseAuth = FirebaseAuth.getInstance();  
    awesomeValidation = new AwesomeValidation(ValidationStyle.BASIC);  
    awesomeValidation.addValidation(this, R.id.email,  
Patterns.EMAIL_ADDRESS, R.string.invalid_mail);  
    awesomeValidation.addValidation(this, R.id.password, "{6,}",  
R.string.invalid_password);  
  
    inicioSesion.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            if (awesomeValidation.validate()) {
```

```

        String email = emailAlumno.getText().toString();
        String password = passwordAlumno.getText().toString();

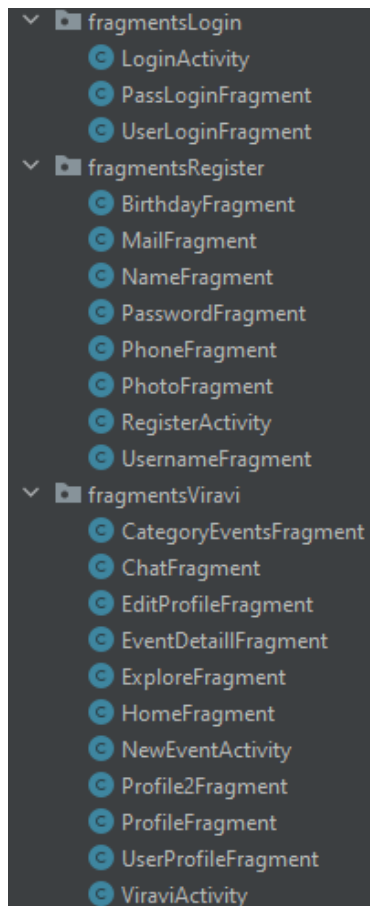
        firebaseAuth.signInWithEmailAndPassword(email,
password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult>
task) {
                if (task.isSuccessful()) {
                    changetoFirebase();
                } else {
                    String errorCode = ((FirebaseAuthException)
task.getException()).getErrorCode();
                    dameToastdeerror(errorCode);
                }
            }
        });
    }

    }
});
}

public void changetoFirebase() {
    Intent actividad = new Intent(MainActivity.this, Firebase.class);
    actividad.putExtra("email", emailAlumno.getText().toString());
    actividad.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_CLEAR_TASK | Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(actividad);
}
}

```

Tras el uso de los fragments, la biblioteca de clases creadas aumentó considerablemente lo que produjo el exceso de complejidad que se mencionaba anteriormente debido a que estas nuevas clases, requerían de mucho código con novedosas funciones para unirse y realizar el traspaso de datos.



Actualmente, el inicio de sesión y el registro de usuario en Firebase, se realizan de la siguiente manera:

```
@Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                            Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        binding = FragmentPhotoBinding.inflate(getLayoutInflater());
        View view = binding.getRoot();
        firebaseAuth = FirebaseAuth.getInstance();
        storage = FirebaseStorage.getInstance();

        binding.btnFoto.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                cargarImagen();
            }
        });
    }
```

```

        return view;
    }

    private void cargarImagen() {

        Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);

        intent.setType("image/");

        startActivityForResult(Intent.createChooser(intent,"Seleccione
la Aplicacion"),10);
    }

    public void onActivityResult(int requestCode, int resultCode,
Intent data){
        super.onActivityResult(requestCode,resultCode,data);
        if(resultCode==RESULT_OK){
            path=data.getData();

            Bundle datosRecuperados = getArguments();
            nombre = datosRecuperados.getString("nombre");
            fecha = datosRecuperados.getString("fecha");
            phone = datosRecuperados.getString("phone");
            mail = datosRecuperados.getString("mail");
            contra = datosRecuperados.getString("pass");
            user = datosRecuperados.getString("user");

            Log.i("path",path.toString());
            binding.btnFoto.setImageURI(path);
            binding.progressBar.setVisibility(View.VISIBLE);

            final StorageReference reference =
storage.getReference().child("profile/"+user);
            reference.putFile(path).addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(UploadTask.TaskSnapshot
taskSnapshot) {

```

```

        reference.getDownloadUrl().addOnSuccessListener(new
OnSuccessListener<Uri>() {
            @Override
            public void onSuccess(Uri uri) {
                ArrayList<Evento> eventosApuntado = null;
                ArrayList<Evento> eventosParticipado =
null;

                ArrayList<Evento> eventosCreado = null;

                binding.progressBar.setVisibility(View.GONE
);

                registrarUsuarioFirebase(user, nombre,
phone, fecha, mail, contra, String.valueOf(uri), 0,
0,eventosApuntado,eventosParticipado,eventosCreado);

            }
        });
    }
});
}

private void registrarUsuarioFirebase(String nombreUsuario, String
nombre, String telefono, String fechaNacimiento, String correo, String
contrasenia, String fotoPerfil, int seguidores, int seguidos,
ArrayList<Evento> eventosApuntado, ArrayList<Evento>
eventosParticipado, ArrayList<Evento> eventosCreados) {
    firebaseAuth.createUserWithEmailAndPassword(correo,contrasenia)
.addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            firebaseAuth.signInWithEmailAndPassword(correo,contrase
nia).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult>
task) {

                    if(task.isSuccessful()){

```



```

        Usuario usuario = new
Usuario(nombreUsuario,nombre,telefono,fechaNacimiento,correo,contraseny
a,fotoPerfil,seguidores,seguidos,eventosApuntado,eventosParticipado,eve
ntosCreados);

        mRootreference.child(usuario.getNombreUsuar
io()).setValue(usuario);

        startActivity(new
Intent(getActivity(),ViraviActivity.class));
    }else{
        Log.i("SignIn",correo+" "+contrasenya);
        String errorcode =
String.valueOf(((FirebaseAuthException) task.getException()));
        Toast.makeText(getContext(),errorcode,Toast
.LENGTH_SHORT).show();
        Log.i("error",errorcode);
    }
}
});
}
});
}
}

```

Y, a continuación, el inicio de sesión:

```

@Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,

        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_pass_login,
container, false);
        btn = view.findViewById(R.id.btnPassLogin);
        passwordUsuario = view.findViewById(R.id.txtPassLogin);
        passwordUsuario.requestFocus();
        passwordUsuario.setText("06112002Ra");

        firebaseAuth= FirebaseAuth.getInstance();
        InputMethodManager im = (InputMethodManager)
getActivity().getSystemService(Context.INPUT_METHOD_SERVICE);
    }
}

```

```

        im.toggleSoftInput(InputMethodManager.SHOW_FORCED,
InputMethodManager.HIDE_IMPLICIT_ONLY);
        passwordUsuario.addTextChangedListener(textWatcher);

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Bundle datosRecuperados = getArguments();
                emailUsuario = datosRecuperados.getString("email");

                firebaseAuth.signInWithEmailAndPassword(emailUsuario,pa
sswordUsuario.getText().toString()).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult>
task) {

                        if(task.isSuccessful()){
                            startActivity(new Intent(getActivity(),
ViraviActivity.class));
                        }else{
                            String errorCode = ((FirebaseAuthException)
task.getException()).getErrorCode();
                            dameToastDeError(errorCode);
                        }
                    }
                });
                im.hideSoftInputFromWindow(passwordUsuario.getWindowTok
en(), 0);
            }
        });
        return view;
    }
}

```

Como puede observarse, el cambio con respecto al inicio de sesión anterior, es grande.

Estas son algunas de clases que se utilizaron para la inserción de datos y la posterior recogida de los mismos.

Otra de las funcionalidades que llevó bastante tiempo fue el chat a tiempo real, que no fue

creado de la forma más convencional existente, sino que es una inserción de “ArrayList” en la base de datos y que es recargada constantemente para dar así la sensación de “chat a tiempo real”, claro está que no posee seguridad, pero este apartado es algo que se ha pensado realizar a futuro, para poder hacer de VIRAVI una aplicación mundial y, sobre todo, segura.

```
public class MessageAdapter extends
RecyclerView.Adapter<MessageAdapter.MessageViewHolder> {
    public List<Message> messageList;

    public MessageAdapter(List<Message> messageList) {
        this.messageList = messageList;
    }

    @Override
    public MessageViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        LayoutInflater inflater =
LayoutInflater.from(parent.getContext());
        MessageCardBinding binding =
MessageCardBinding.inflate(inflater, parent, false);
        return new MessageViewHolder(binding);
    }

    @Override
    public void onBindViewHolder(@NonNull MessageViewHolder holder, int
position) {
        Message message = messageList.get(position);
        holder.bind(message);
    }

    @Override
    public int getItemCount() {
        return messageList != null ? messageList.size() : 0;
    }

    public class MessageViewHolder extends RecyclerView.ViewHolder {
        private MessageCardBinding binding;
```

```

public MessageViewHolder(MessageCardBinding binding) {
    super(binding.getRoot());
    this.binding = binding;
}

public void bind(Message message) {
    if (message.getText().equals("!Bienvenidos!")) {
        binding.welcome.getRoot().setVisibility(View.VISIBLE);
        binding.sender.getRoot().setVisibility(View.GONE);
        binding.receiver.getRoot().setVisibility(View.GONE);

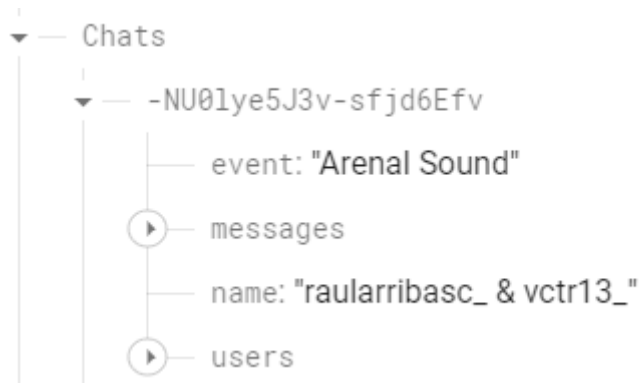
        binding.welcome.tvText.setText(message.getText());
    } else {
        if (message.getOwner()) {
            binding.sender.getRoot().setVisibility(View.VISIBLE);
            binding.receiver.getRoot().setVisibility(View.GONE);
            binding.welcome.getRoot().setVisibility(View.GONE);

            binding.sender.tvText.setText(message.getText());
            binding.sender.tvTime.setText(message.getTimestamp());
        } else {
            binding.sender.getRoot().setVisibility(View.GONE);
            binding.receiver.getRoot().setVisibility(View.VISIBLE);
            binding.welcome.getRoot().setVisibility(View.GONE);

            binding.receiver.tvText.setText(message.getText());
            binding.receiver.tvTime.setText(message.getTimestamp());
        }
    }
}
}
}
}
}

```

En esta parte se muestra cómo funcionan los chats internamente y como ofrecen sensación de conversación a tiempo real.



7. Conclusiones

VIRAVI es fruto de la investigación, de la ayuda de profesores, del sacrificio grupal, del tiempo empleado, de la imaginación y por su puesto del trabajo en equipo. Esta red social ha enseñado que, con esfuerzo todo es posible en el ámbito de la programación, que todo lo propuesto es posible si se intenta, no siempre todo sale como se desea o no siempre es posible cumplir los plazos de tiempo, a veces aparece el estrés, la ansiedad y la amargura, pero siempre se puede solucionar con ayuda de los demás. El trabajo en equipo, como ya se ha mencionado anteriormente, ha sido el factor más importante y sin el cual, esto no hubiese sido posible.

Por su puesto, es muy importante saber programar y tener nociones básicas de cómo funcionan ciertas aplicaciones o como se debe realizar alguna validación de datos, pero todo puede aprenderse, nadie nace sabiendo y la actitud positiva, discutir lo menos posible y ser compañeros ha sido lo más importante y lo que más enorgullece.

Para concluir con todo lo anterior, podría comentarse, lo mucho que se ha aprendido con este proyecto, el cariño que lleva implícito y el trabajo dedicado, por su puesto, esta aplicación no está terminada, aún se pueden y se deben añadir funcionalidades extras, además de mejorar las ya existentes y aplicar medidas de seguridad de datos, tales como la encriptación de contraseñas, en caso de pretender llegar a un público mayor.

A pesar de todo esto, se está muy orgulloso del trabajo realizado y de todo lo que ha conllevado y conllevará VIRAVI.

8. Bibliografía.

Aloompa. (2023, 29 marzo). *Aloompa / Live Event Technology*. Disponible en <https://aloompa.com/>.

Figma. (s. f.). Figma. Disponible en <https://www.figma.com/files/recentfuid=1214607195416517342>

Meetup. (2023, 2 mayo). *Meetup: somos lo que hacemos*. Disponible en <https://www.meetup.com/es-ES/>.

Whova (2023): *Award-winning Event Apps and Event Management Software* / Whova. Disponible en <https://whova.com/>