

# INSTADAM

JOSEP FONFREDÀ  
MARC RIERA

# 1. Introducción

¡Bienvenido/a a INSTADAM! Esta aplicación te brinda una experiencia única para compartir y disfrutar de fotos. Este manual te guiará a través de los pasos esenciales para aprovechar al máximo la aplicación.

```
class MyApp extends StatelessWidget {  
  const MyApp({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return MultiProvider(  
      providers: [  
        ChangeNotifierProvider(create: (_) => UserProvider(),),  
      ],  
      child: MaterialApp(  
        debugShowCheckedModeBanner: false,  
        title: 'Instagram Clone',  
        theme: ThemeData.dark().copyWith(  
          scaffoldBackgroundColor: mobileBackgroundColor,  
        ),  
        home: StreamBuilder(  
          stream: FirebaseAuth.instance.authStateChanges(),  
          builder: (context, snapshot) {  
            if (snapshot.connectionState == ConnectionState.active) {  
              // Checking if the snapshot has any data or not  
              if (snapshot.hasData) {  
                // if snapshot has data which means user is logged in then we check the width of screen a  
                return const ResponsiveLayout(  
                  mobileScreenLayout: MobileScreenLayout(),  
                  webScreenLayout: WebScreenLayout(),  
                );  
              }  
            }  
            return const ResponsiveLayout(  
              mobileScreenLayout: MobileScreenLayout(),  
              webScreenLayout: WebScreenLayout(),  
            );  
          },  
        ),  
      ),  
    );  
  }  
}
```

## 2. Registro

```

1  import ...
13
14  class SignupScreen extends StatefulWidget {
15      const SignupScreen({Key? key}) : super(key: key);
16
17      @override
18      State<SignupScreen> createState() => _SignupScreenState();
19  }
20
21  class _SignupScreenState extends State<SignupScreen> {
22      final TextEditingController _usernameController = TextEditingController();
23      final TextEditingController _emailController = TextEditingController();
24      final TextEditingController _passwordController = TextEditingController();
25      final TextEditingController _bioController = TextEditingController();
26      bool _isLoading = false;
27      Uint8List? _image;
28
29      @override
30      void dispose() {
31          super.dispose();
32          _emailController.dispose();
33          _passwordController.dispose();
34          _usernameController.dispose();
35      }
36
37      void signUpUser() async {

```

```

38          // set loading to true
39          setState(() {
40              _isLoading = true;
41          });
42
43          // signup user using our authmethodds
44          String res = await AuthMethods().signUpUser(
45              email: _emailController.text,
46              password: _passwordController.text,
47              username: _usernameController.text,
48              bio: _bioController.text,
49              file: _image!);
50          // if string returned is sucess, user has been created
51          if (res == "success") {
52              setState(() {
53                  _isLoading = false;
54              });
55              // navigate to the home screen
56              if (context.mounted) {
57                  Navigator.of(context).pushReplacement(
58                      MaterialPageRoute(
59                          builder: (context) => const ResponsiveLayout(
60                              mobileScreenLayout: MobileScreenLayout(),
61                              webScreenLayout: WebScreenLayout()

```

```

62                      ),
63                      ),
64                  );
65              }
66          } else {
67              setState(() {
68                  _isLoading = false;
69              });
70              // show the error
71              if (context.mounted) {
72                  showSnackBar(context, res);
73              }
74          }
75      }
76
77      selectImage() async {
78          Uint8List im = await pickImage(ImageSource.gallery);
79          // set state because we need to display the image we selected on the circle avatar
80          setState(() {
81              _image = im;
82          });
83      }
84
85      @override
86      Widget build(BuildContext context) {

```

```

88     resizeToAvoidBottomInset: false,
89     body: SafeArea(
90       child: Container(
91         padding: const EdgeInsets.symmetric(horizontal: 32),
92         width: double.infinity,
93         child: Column(
94           crossAxisAlignment: CrossAxisAlignment.center,
95           children: [
96             Flexible(
97               flex: 2,
98               child: Container(),
99             ),
100             SvgPicture.asset(
101               'assets/ic_instagram.svg',
102               color: primaryColor,
103               height: 64,
104             ),
105             const SizedBox(
106               height: 64,
107             ),
108             Stack(
109               children: [
110                 image != null

```

```

111                 ? CircleAvatar(
112                   radius: 64,
113                   backgroundImage: MemoryImage(_image!),
114                   backgroundColor: Colors.red,
115                 )
116                 : const CircleAvatar(
117                   radius: 64,
118                   backgroundImage: NetworkImage(
119                     'https://i.stack.imgur.com/l60Hf.png'),
120                   backgroundColor: Colors.red,
121                 ),
122             Positioned(
123               bottom: -10,
124               left: 80,
125               child: IconButton(
126                 onPressed: selectImage,
127                 icon: const Icon(Icons.add_a_photo),
128               ),
129             )
130           ],
131         ),
132         const SizedBox(
133           height: 24,
134         ),
135       TextFieldInput(

```

```

135       TextFieldInput(
136         hintText: 'Enter your username',
137         textInputType: TextInputType.text,
138         textEditingController: _usernameController,
139       ),
140       const SizedBox(
141         height: 24,
142       ),
143       TextFieldInput(
144         hintText: 'Enter your email',
145         textInputType: TextInputType.emailAddress,
146         textEditingController: _emailController,
147       ),
148       const SizedBox(
149         height: 24,
150       ),
151       TextFieldInput(
152         hintText: 'Enter your password',
153         textInputType: TextInputType.text,
154         textEditingController: _passwordController,
155         isPass: true,
156       ),
157       const SizedBox(
158         height: 24,
159       ),

```

```

158     ),
159   ),
160   TextFieldInput(
161     hintText: 'Enter your bio',
162     textInputType: TextInputType.text,
163     textEditingController: _bioController,
164   ),
165   const SizedBox(
166     height: 24,
167   ),
168   InkWell(
169     onTap: signUpUser,
170     child: Container(
171       width: double.infinity,
172       alignment: Alignment.center,
173       padding: const EdgeInsets.symmetric(vertical: 12),
174       decoration: const ShapeDecoration(
175         shape: RoundedRectangleBorder(
176           borderRadius: BorderRadius.all(Radius.circular(4)),
177         ),
178         color: blueColor,
179       ),
180       child: !_isLoading
181         ? const Text(
182             'Sign up',
183           )

```

```

184       : const CircularProgressIndicator(
185         color: primaryColor,
186       ),
187     ),
188   ),
189   const SizedBox(
190     height: 12,
191   ),
192   Flexible(
193     flex: 2,
194     child: Container(),
195   ),
196   Row(
197     mainAxisAlignment: MainAxisAlignment.center,
198     children: [
199       Container(
200         padding: const EdgeInsets.symmetric(vertical: 8),
201         child: const Text(
202           'Already have an account?',
203         ),
204       ),
205       GestureDetector(
206         onTap: () => Navigator.of(context).push(
207           MaterialPageRoute(
208             builder: (context) => const LoginScreen()

```

```

204         ),
205         GestureDetector(
206             onTap: () => Navigator.of(context).push(
207                 MaterialPageRoute(
208                     builder: (context) => const LoginScreen(),
209                 ),
210             ),
211         child: Container(
212             padding: const EdgeInsets.symmetric(vertical: 8),
213             child: const Text(
214                 'Login.',
215                 style: TextStyle(
216                     fontWeight: FontWeight.bold,
217                 ),
218             ),
219         ),
220     ),
221     ],
222 ),
223 ],
224 ),
225 ),
226 ),
227 );
228 }
229 }

```

## 2.1 Crear una Cuenta

Para comenzar a utilizar INSTADAM, sigue estos sencillos pasos:

- Descarga la aplicación desde la tienda de aplicaciones.
- Abre la aplicación e inicia el proceso de registro.
- Completa la información requerida, incluyendo un correo electrónico válido y una contraseña segura.
- Verifica tu cuenta a través del enlace enviado a tu correo electrónico.

## 2.2 Iniciar Sesión

Si ya tienes una cuenta, sigue estos pasos para iniciar sesión:

- Abre la aplicación.
- Ingresa tu dirección de correo electrónico y contraseña.
- Haz clic en "Iniciar Sesión".

```

1  import ...
2
13  class LoginScreen extends StatefulWidget {
14      const LoginScreen({Key? key}) : super(key: key);
15
16      @override
17      State<LoginScreen> createState() => _LoginScreenState();
18  }
19
20  class _LoginScreenState extends State<LoginScreen> {
21      final TextEditingController _emailController = TextEditingController();
22      final TextEditingController _passwordController = TextEditingController();
23      bool _isLoading = false;
24  }

```

```

41 |         MaterialPageRoute(
42 |           builder: (context) => const ResponsiveLayout(
43 |             mobileScreenLayout: MobileScreenLayout(),
44 |             webScreenLayout: WebScreenLayout(),
45 |           ),
46 |         ),
47 |         (route) => false);
48 |
49 |       setState(() {
50 |         _isLoading = false;
51 |       });
52 |     }
53 |   } else {
54 |     setState(() {
55 |       _isLoading = false;
56 |     });
57 |     if (context.mounted) {
58 |       showSnackBar(context, res);
59 |     }
60 |   }
61 | }
62 |
63 | @override
64 | Widget build(BuildContext context) {
65 |   return Scaffold(
66 |     resizeToAvoidBottomInset: false

```

```

return Scaffold(
  resizeToAvoidBottomInset: false,
  body: SafeArea(
    child: Container(
      padding: MediaQuery.of(context).size.width > webScreenSize
        ? EdgeInsets.symmetric(
            horizontal: MediaQuery.of(context).size.width / 3
          ) : const EdgeInsets.symmetric(horizontal: 32),
      width: double.infinity,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
          Flexible(
            flex: 2,
            child: Container(),
          ),
          SvgPicture.asset(
            'assets/ic_instagram.svg',
            color: primaryColor,
            height: 64,
          ),
          const SizedBox(
            height: 64,
          ),
          TextFieldInput(
            hintText: 'Enter your email'

```

```

            color: primaryColor,
            height: 64,
          ),
          const SizedBox(
            height: 64,
          ),
          TextFieldInput(
            hintText: 'Enter your email',
            keyboardType: TextInputType.emailAddress,
            textEditingController: _emailController,
          ),
          const SizedBox(
            height: 24,
          ),
          TextFieldInput(
            hintText: 'Enter your password',
            keyboardType: TextInputType.text,
            textEditingController: _passwordController,
            isPass: true,
          ),
          const SizedBox(
            height: 24,
          ),
          InkWell(
            onTap: loginUser,
            child: Container(

```

```
GestureDetector(
  onTap: () => Navigator.of(context).push(
    MaterialPageRoute(
      builder: (context) => const SignupScreen(),
    ),
  ),
  child: Container(
    padding: const EdgeInsets.symmetric(vertical: 8),
    child: const Text(
      'Signup.',
      style: TextStyle(
        fontWeight: FontWeight.bold,
      ),
    ),
  ),
),
),
),
),
),
),
),
),
);
}
```

### 3. Subir Fotos

Compartir tus momentos favoritos es fácil. Sigue estos pasos para subir fotos a tu perfil:

En la página principal, haz clic en el ícono de "+", generalmente ubicado en la parte inferior de la pantalla.

Selecciona la foto que deseas compartir desde tu galería.

Personaliza tu foto con filtros y ajustes disponibles.

Escribe una descripción creativa para acompañar tu foto.

Publica tu foto para que tus seguidores la vean.



```

1  import ...
10
11  class AddPostScreen extends StatefulWidget {
12    const AddPostScreen({Key? key}) : super(key: key);
13
14    @override
15    _AddPostScreenState createState() => _AddPostScreenState();
16  }
17
18  class _AddPostScreenState extends State<AddPostScreen> {
19    Uint8List? _file;
20    bool isLoading = false;
21    final TextEditingController _descriptionController = TextEditingController();
22
23    _selectImage(BuildContext parentContext) async {
24      return showDialog(
25        context: parentContext,
26        builder: (BuildContext context) {
27          return SimpleDialog(
28            title: const Text('Create a Post'),
29            children: <Widget>[
30              SimpleDialogOption(
31                padding: const EdgeInsets.all(20),
32                child: const Text('Take a photo'),
33                onPressed: () async {
34                  Navigator.pop(context);

```

```

35                  Uint8List file = await pickImage(ImageSource.camera);
36                  setState(() {
37                    _file = file;
38                  });
39                },
40              SimpleDialogOption(
41                padding: const EdgeInsets.all(20),
42                child: const Text('Choose from Gallery'),
43                onPressed: () async {
44                  Navigator.of(context).pop();
45                  Uint8List file = await pickImage(ImageSource.gallery);
46                  setState(() {
47                    _file = file;
48                  });
49                },
50              SimpleDialogOption(
51                padding: const EdgeInsets.all(20),
52                child: const Text("Cancel"),
53                onPressed: () {
54                  Navigator.pop(context);
55                },
56            ],
57          );
58        },
59      );
60    }

```

```

63  void postImage(String uid, String username, String profImage) async {
64    setState(() {
65      isLoading = true;
66    });
67    // start the loading
68    try {
69      // upload to storage and db
70      String res = await FirestoreMethods().uploadPost(
71        _descriptionController.text,
72        _file!,
73        uid,
74        username,
75        profImage,
76      );
77      if (res == "success") {
78        setState(() {
79          isLoading = false;
80        });
81        if (context.mounted) {
82          showSnackBar(
83            context,
84            'Posted!',
85          );
86        }
87      }
88      cleanImage();

```

```

88     } else {
89         if (context.mounted) {
90             showSnackBar(context, res);
91         }
92     }
93 } catch (err) {
94     setState(() {
95         isLoading = false;
96     });
97     showSnackBar(
98         context,
99         err.toString(),
100     );
101 }
102 }
103
104 void clearImage() {
105     setState(() {
106         _file = null;
107     });
108 }
109
110 @override
111 void dispose() {
112     super.dispose();

```

```

111 void dispose() {
112     super.dispose();
113     _descriptionController.dispose();
114 }
115
116 @override
117 Widget build(BuildContext context) {
118     final UserProvider userProvider = Provider.of<UserProvider>(context);
119
120     return _file == null
121         ? Center(
122             child: IconButton(
123                 icon: const Icon(
124                     Icons.upload,
125                 ),
126                 onPressed: () => _selectImage(context),
127             ),
128         )
129         : Scaffold(
130             appBar: AppBar(
131                 backgroundColor: mobileBackgroundColor,
132                 leading: IconButton(
133                     icon: const Icon(Icons.arrow_back),
134                     onPressed: clearImage,
135                 ),
136             title: const Text(

```

```

138     ),
139     centerTitle: false,
140     actions: <Widget>[
141       TextButton(
142         onPressed: () => postImage(
143           userProvider.getUser.uid,
144           userProvider.getUser.username,
145           userProvider.getUser.photoUrl,
146         ),
147         child: const Text(
148           "Post",
149           style: TextStyle(
150             color: Colors.blueAccent,
151             fontWeight: FontWeight.bold,
152             fontSize: 16.0),
153         ),
154       ),
155     ],
156   ),
157   // POST FORM
158   body: Column(
159     children: <Widget>[
160       isLoading
161         ? const LinearProgressIndicator()
162         : const Padding(padding: EdgeInsets.only(top: 0.0)),
163       const Divider()

```

```

163     const Divider(),
164     Row(
165       mainAxisAlignment: MainAxisAlignment.spaceAround,
166       crossAxisAlignment: CrossAxisAlignment.start,
167       children: <Widget>[
168         CircleAvatar(
169           backgroundImage: NetworkImage(
170             userProvider.getUser.photoUrl,
171           ),
172         ),
173         SizedBox(
174           width: MediaQuery.of(context).size.width * 0.3,
175           child: TextField(
176             controller: _descriptionController,
177             decoration: const InputDecoration(
178               hintText: "Write a caption...",
179               border: InputBorder.none,
180               maxLines: 8,
181             ),
182           ),
183         SizedBox(
184           height: 45.0,
185           width: 45.0,
186           child: AspectRatio(
187             aspectRatio: 487 / 451,
188             child: Container(

```

```

188             height: 45.0,
189             width: 45.0,
190             child: AspectRatio(
191               aspectRatio: 487 / 451,
192               child: Container(
193                 decoration: BoxDecoration(
194                   image: DecorationImage(
195                     fit: BoxFit.fill,
196                     alignment: FractionalOffset.topCenter,
197                     image: MemoryImage(_file!),
198                   )),
199               ),
200             ),
201           ),
202         ),
203       ],
204     ),
205   );

```

## 4. Comentar Fotos

La interacción es clave en INSTADAM. Comenta las fotos de tus amigos o personas que sigues:

Abre la foto en la que deseas comentar.

Debajo de la foto, encontrarás un campo de comentarios. Haz clic en "Agregar un comentario".

Escribe tu comentario y presiona "Publicar" para compartir tus pensamientos.

```
1  import ...
10
11  class CommentsScreen extends StatefulWidget {
12    final postId;
13    const CommentsScreen({Key? key, required this.postId}) : super(key: key);
14
15    @override
16    _CommentsScreenState createState() => _CommentsScreenState();
17  }
18
19  class _CommentsScreenState extends State<CommentsScreen> {
20    final TextEditingController commentEditingController =
21      TextEditingController();
22
23    void postComment(String uid, String name, String profilePic) async {
24      try {
25        String res = await FirestoreMethods().postComment(
26          widget.postId,
27          commentEditingController.text,
28          uid,
29          name,
30          profilePic,
31        );
32
33        if (res != 'success') {
34          if (context.mounted) showSnackBar(context, res);
```

```
34          if (context.mounted) showSnackBar(context, res);
35        }
36        setState(() {
37          commentEditingController.text = "";
38        });
39      } catch (err) {
40        showSnackBar(
41          context,
42          err.toString(),
43        );
44      }
45    }
46
47    @override
48    Widget build(BuildContext context) {
49      final User user = Provider.of<UserProvider>(context).getUser;
50
51      return Scaffold(
52        appBar: AppBar(
53          backgroundColor: mobileBackgroundColor,
54          title: const Text(
55            'Comments',
56          ),
57          centerTitle: false,
58        ),
```

```

59 body: StreamBuilder(
60   stream: FirebaseFirestore.instance
61     .collection('posts')
62     .doc(widget.postId)
63     .collection('comments')
64     .snapshots(),
65   builder: (context,
66     AsyncSnapshot<QuerySnapshot<Map<String, dynamic>>> snapshot) {
67     if (snapshot.connectionState == ConnectionState.waiting) {
68       return const Center(
69         child: CircularProgressIndicator(),
70       );
71     }
72
73     return ListView.builder(
74       itemCount: snapshot.data!.docs.length,
75       itemBuilder: (ctx, index) => CommentCard(
76         snap: snapshot.data!.docs[index],
77       ),
78     );
79   },
80 ),
81 // text input
82 bottomNavigationBar: SafeArea(

```

```

82   bottomNavigationBar: SafeArea(
83     child: Container(
84       height: kToolbarHeight,
85       margin:
86         EdgeInsets.only(bottom: MediaQuery.of(context).viewInsets.bottom),
87       padding: const EdgeInsets.only(left: 16, right: 8),
88       child: Row(
89         children: [
90           CircleAvatar(
91             backgroundImage: NetworkImage(user.photoUrl),
92             radius: 18,
93           ),
94           Expanded(
95             child: Padding(
96               padding: const EdgeInsets.only(left: 16, right: 8),
97               child: TextField(
98                 controller: commentEditingController,
99                 decoration: InputDecoration(
100                   hintText: 'Comment as ${user.username}',
101                   border: InputBorder.none,
102                 ),
103               ),
104             ),
105           ),
106         ],
107       ),
108     ),
109   ),
110   InkWell(

```

```

110   ),
111   InkWell(
112     onTap: () => postComment(
113       user.vid,
114       user.username,
115       user.photoUrl,
116     ),
117     child: Container(
118       padding:
119         const EdgeInsets.symmetric(vertical: 8, horizontal: 8),
120       child: const Text(
121         'Post',
122         style: TextStyle(color: Colors.blue),
123       ),
124     ),
125   ),
126 ],
127 ),
128 ),
129 );
130 }
131 }

```

## 5. Soporte y Contacto

Si encuentras algún problema o tienes preguntas, no dudes en contactar a nuestro equipo de soporte:

- Correo Electrónico: [support@INSTADAM.com](mailto:support@INSTADAM.com)
- Centro de Ayuda en la aplicación

¡Disfruta compartiendo tus momentos en INSTADAM ! Estamos emocionados de tenerte a bordo.