



INGENIERÍA DE
SOFTWARE

APLICACIONES MÓVILES

**FIREBASE
AUTHENTICATION
CON FACEBOOK**



DOCENTE: CRISTHIAN ZAMBRANO, PHD

EQUIPO



JOSE
LOZANO

SUGGAR DADY



ANDY
SANCHEZ

EL QUE PONE LA COLA



IA VOV
JEREMY
GAIBOR

INGENIERO EN
PROMPT



JHINSON
AUCATOMA

LA CABRITA



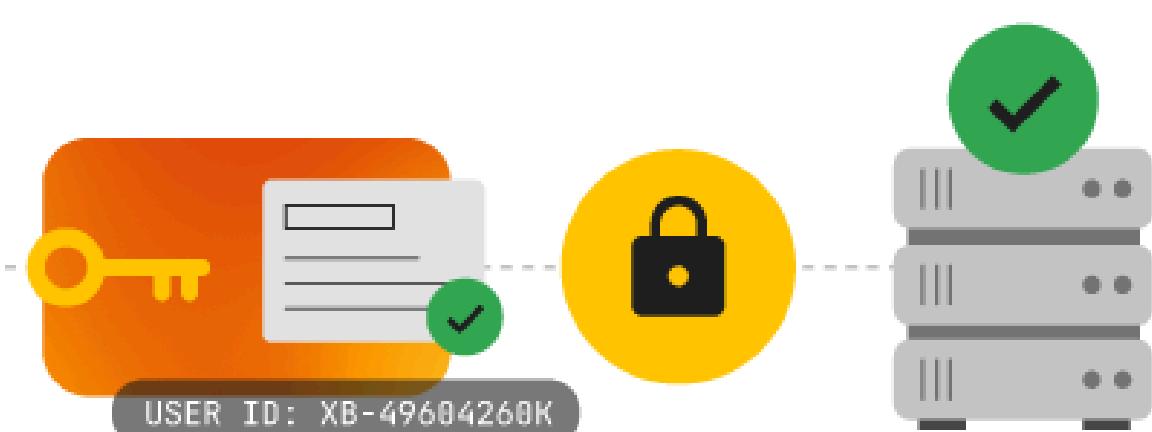
CRISTHIAN
PACHECO

INGENIERO EN
ACCIÓN

¿QUÉ ES FIREBASE AUTHENTICATION?

La autenticación de Firebase es una característica de la autenticación de usuario que proporciona Firebase como sus servicios de backend.

Este es un sistema de autenticación basado en token que proporciona una fácil integración con la mayoría de las plataformas.



USO DE CUENTAS EXISTENTES COMO ALTERNATIVA AL REGISTRO TRADICIONAL

En lugar de que el usuario cree una cuenta nueva con correo y contraseña, Firebase permite autenticarse mediante proveedores externos ya existentes.



#1

Mayor rapidez en el acceso

#2

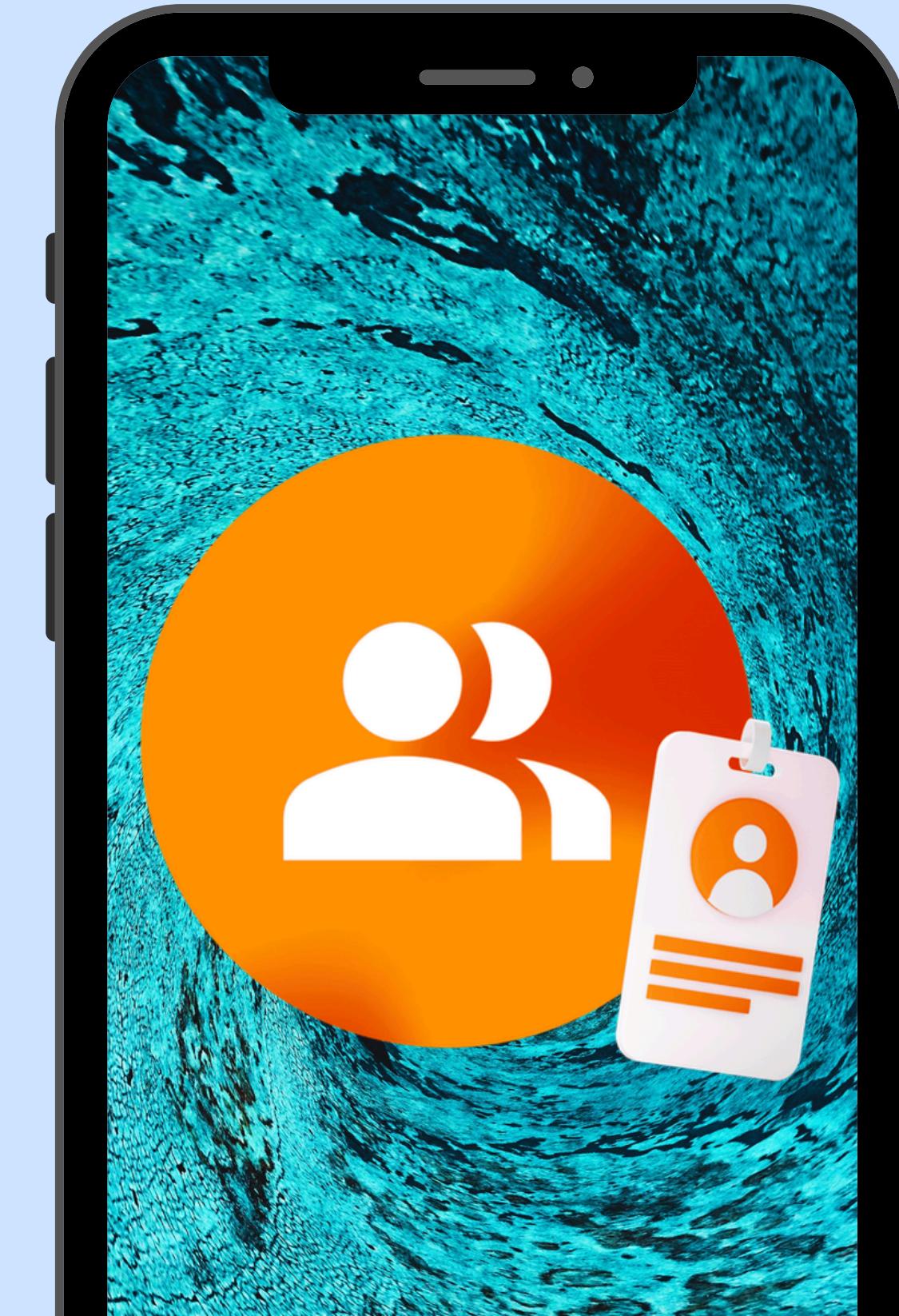
Mejor experiencia de usuario

#3

No se necesitan nuevas
contraseñas

#4

Mayor seguridad (OAuth 2.0)

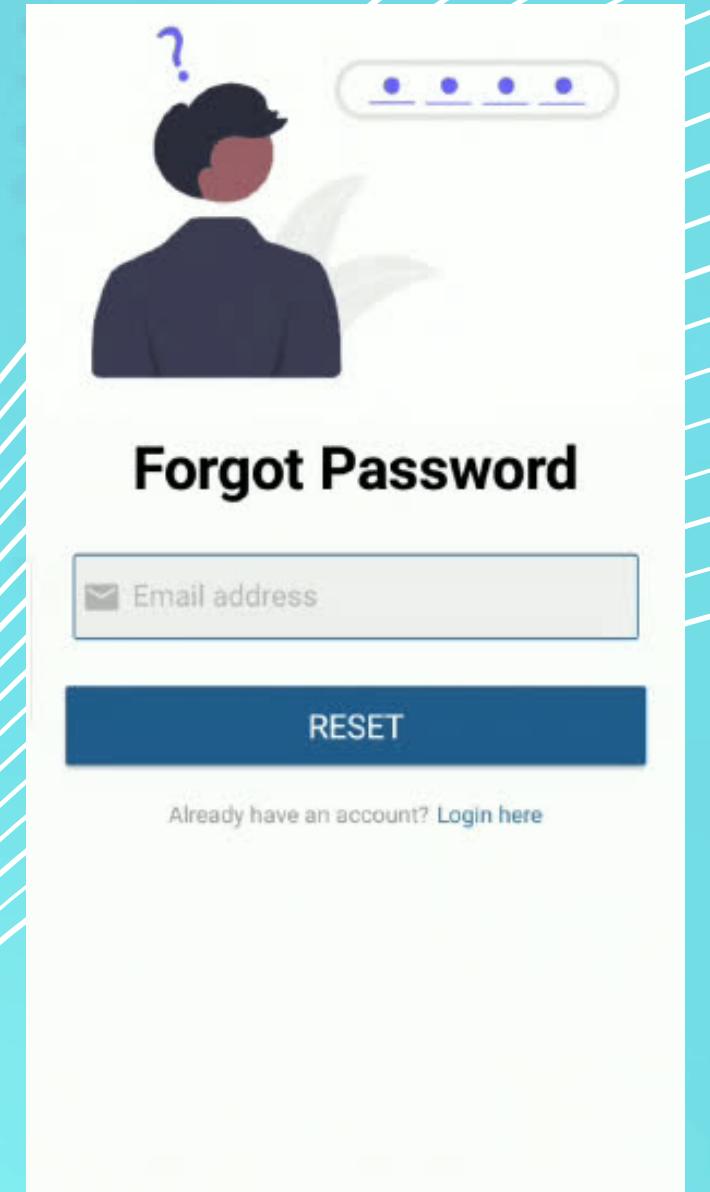


USO DE FACEBOOK COMO PROVEEDOR DE IDENTIDAD

Permite entrar a una aplicación utilizando una cuenta que la persona ya tiene en Facebook, sin necesidad de crear una cuenta nueva dentro de la aplicación. Facebook verifica que el usuario es auténtico y comunica a la aplicación que puede permitir el acceso. La aplicación no controla la cuenta ni conoce la contraseña; únicamente confía en la confirmación que recibe.

Implica algunas características como:

- El usuario no necesita recordar otra contraseña diferente.
- La cuenta usada es la misma que emplea fuera de la aplicación.
- El acceso puede mantenerse guardado para no iniciar sesión cada vez.



Inicia sesión o regístrate en un momento

Usa tu correo electrónico u otro servicio para acceder a Canva gratis.

Usar Google
 Usar Facebook
 Usar un correo electrónico

Acceder de otra forma

Al continuar, aceptas los [Términos y condiciones de uso](#) de Canva. Consulta nuestra [Política de privacidad](#).

Registrarme con el correo del trabajo



SEGURIDAD Y PRIVACIDAD EN ESTE MÉTODO DE ACCESO

Este sistema busca proteger la información personal durante el inicio de sesión. La contraseña de la cuenta externa nunca se entrega a la aplicación, lo que evita que quede almacenada en lugares inseguros. Además, el usuario tiene control sobre qué datos permite compartir antes de completar el acceso.

Para comprender mejor la protección que ofrece este método:

- El usuario puede cancelar el proceso en cualquier momento antes de aceptar.
- Solo se solicitan datos mínimos necesarios para identificar a la persona.
- El acceso puede revocarse posteriormente desde la propia cuenta externa.



FACEBOOK LOGIN VS OTROS MÉTODOS EN FIREBASE

FACEBOOK LOGIN

01

- Usa cuenta existente
- No requiere nueva contraseña
- Menos pasos que el registro tradicional

GOOGLE SIGN-IN

02

- Inicio rápido
- No crea contraseña nueva
- Reduce fricción frente al registro tradicional

GITHUB / TWITTER

03

- Cuenta existente
- Acceso ágil
- Más simple que crear cuenta manual

CORREO + CONTRASEÑA

04

- Crear cuenta manual
- Requiere recordar contraseña
- Más pasos y mayor abandono

TELÉFONO (SMS)

05

- No usa red social
- Requiere código OTP
- Dependiente del número móvil

Ejemplos reales del uso de Facebook

Login

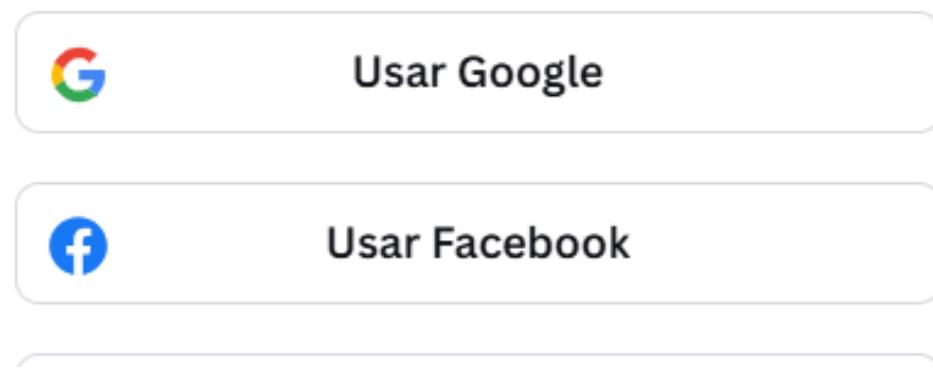
1 Duolingo



2 Canva

< Inicia sesión o regístrate en segundos

Para acceder a Canva gratis, usa tu correo u otro servicio.



3 TikTok



CÓDIGO



```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    EdgeToEdge.enable(this);  
    setContentView(R.layout.activity_main);  
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {  
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());  
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
        return insets;  
    });  
  
    mAuth = FirebaseAuth.getInstance();  
    mCallbackManager = CallbackManager.Factory.create();  
  
    btnLoginFacebook = findViewById(R.id.btnLogin);  
    btnLogout = findViewById(R.id.btnLogout);  
    userInfoTextView = findViewById(R.id.userInfoTextView);  
  
    btnLogout.setOnClickListener(v -> signOut());  
  
    LoginManager.getInstance().registerCallback(mCallbackManager,  
        new FacebookCallback<LoginResult>() {  
            @Override  
            public void onSuccess(LoginResult loginResult) {  
                Log.d(TAG, "facebook:onSuccess:" + loginResult);  
                handleFacebookAccessToken(loginResult.getAccessToken());  
            }  
  
            @Override  
            public void onCancel() {  
                Log.d(TAG, "facebook:onCancel");  
            }  
  
            @Override  
            public void onError(@NonNull FacebookException e) {  
                Log.d(TAG, "facebook:onError", e);  
            }  
        });  
}
```



```
public void login(View view) {  
    login_facebook();  
}
```



```
private void login_facebook() {  
    LoginManager.getInstance().logInWithReadPermissions(this, Arrays.asList("email",  
    "public_profile"));  
}
```



```
private void signOut() {
    if (AccessToken.getCurrentAccessToken() != null) {
        new GraphRequest(AccessToken.getCurrentAccessToken(), "/me/permissions", null, HttpMethod.DELETE, response -> {
            mAuth.signOut();
            LoginManager.getInstance().logOut();
            updateUI(null);
        }).executeAsync();
    } else {
        mAuth.signOut();
        LoginManager.getInstance().logOut();
        updateUI(null);
    }
}
```



```
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
    mCallbackManager.onActivityResult(requestCode, resultCode, data);  
}
```



```
@Override  
public void onStart() {  
    super.onStart();  
    FirebaseUser currentUser = mAuth.getCurrentUser();  
    updateUI(currentUser);  
}
```



```
private void handleFacebookAccessToken(AccessToken token) {
    Log.d(TAG, "handleFacebookAccessToken:" + token);

    AuthCredential credential = FacebookAuthProvider.getCredential(token.getToken());
    mAuth.signInWithCredential(credential)
        .addOnCompleteListener(this, (Task<AuthResult> task) -> {
            if (task.isSuccessful()) {
                Log.d(TAG, "signInWithCredential:success");
                FirebaseUser user = mAuth.getCurrentUser();
                updateUI(user);
            } else {
                Log.w(TAG, "signInWithCredential:failure", task.getException());
                Toast.makeText(MainActivity.this, "Authentication failed.",
                        Toast.LENGTH_SHORT).show();
                updateUI(null);
            }
        });
}
```



```
private void updateUI(FirebaseUser user) {  
    if (user != null) {  
        btnLoginFacebook.setVisibility(View.GONE);  
        btnLogout.setVisibility(View.VISIBLE);  
        String name = user.getDisplayName();  
        String email = user.getEmail();  
        userInfoTextView.setText("Bienvenido, " + name + "\n" + email);  
  
    } else {  
        btnLoginFacebook.setVisibility(View.VISIBLE);  
        btnLogout.setVisibility(View.GONE);  
        userInfoTextView.setText("");  
    }  
}
```

GRACIAS

Me ayudan mucho con un café