



Firestore



Rodrigo Sicarelli

br.linkedin.com/in/rodrigodicarelli

[@rsicarelli](https://twitter.com/rsicarelli)

github.com/rsicarelli



Isabela Terribili

br.linkedin.com/in/isabela-cristina-terribili-baro

[@isabelaterribili](https://twitter.com/isabelaterribili)

[@github.com/isabelaterribili](https://github.com/isabelaterribili)

**Acompanhe os links no
Twitter!**

@rsicarelli @IsabelaTerribil

#AndroidDevConference

Agenda

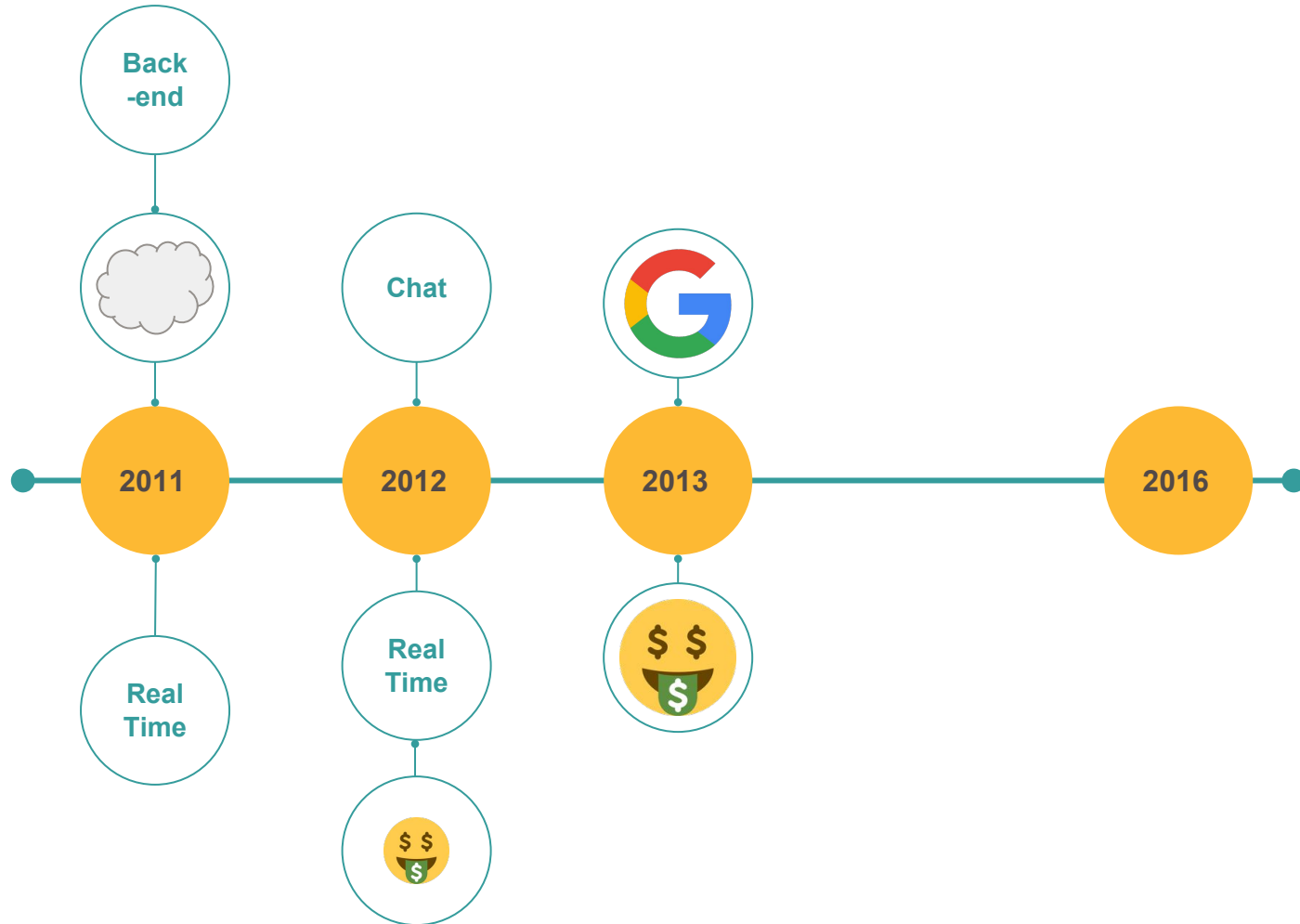
- Fire o que?
- Linha tempo;
- Firebase nos dias de hoje;
- Pilares do Firebase.
- Analytics;
- Considerações finais;
- Nos pergunte!

"Creating a successful app is still too hard"

James Tamplin, Google I/O 2016



Fornecer serviços de back-end para mobile
(Android, iOS e Web) sem que haja necessidade de
mantê-lo.





E finalmente, olá
Firebase!



*Tag Manager, Google Analytics, Google Cloud
Messaging, Play Services ...*

Bem-vindo ao Firebase! Comece aqui.



Adicionar o
Firebase ao seu
aplicativo iOS



Adicionar o
Firebase ao seu
aplicativo
Android

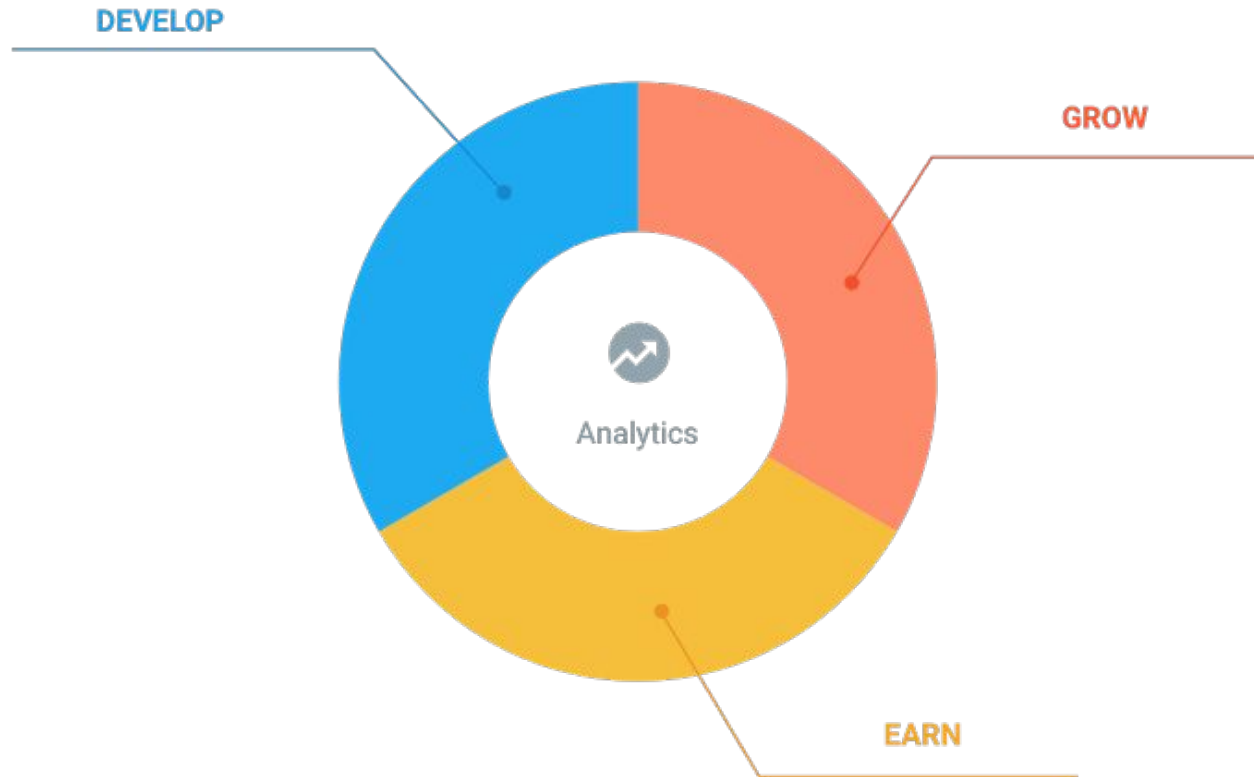


Adicionar o
Firebase ao seu
app da Web

Mobile-first: pensando em mobile, mas não exclusivo!



Pilares



Para nós, desenvolvedores

DEVELOP



Realtime Database



Authentication



Cloud Messaging



Storage



Hosting



Remote Config



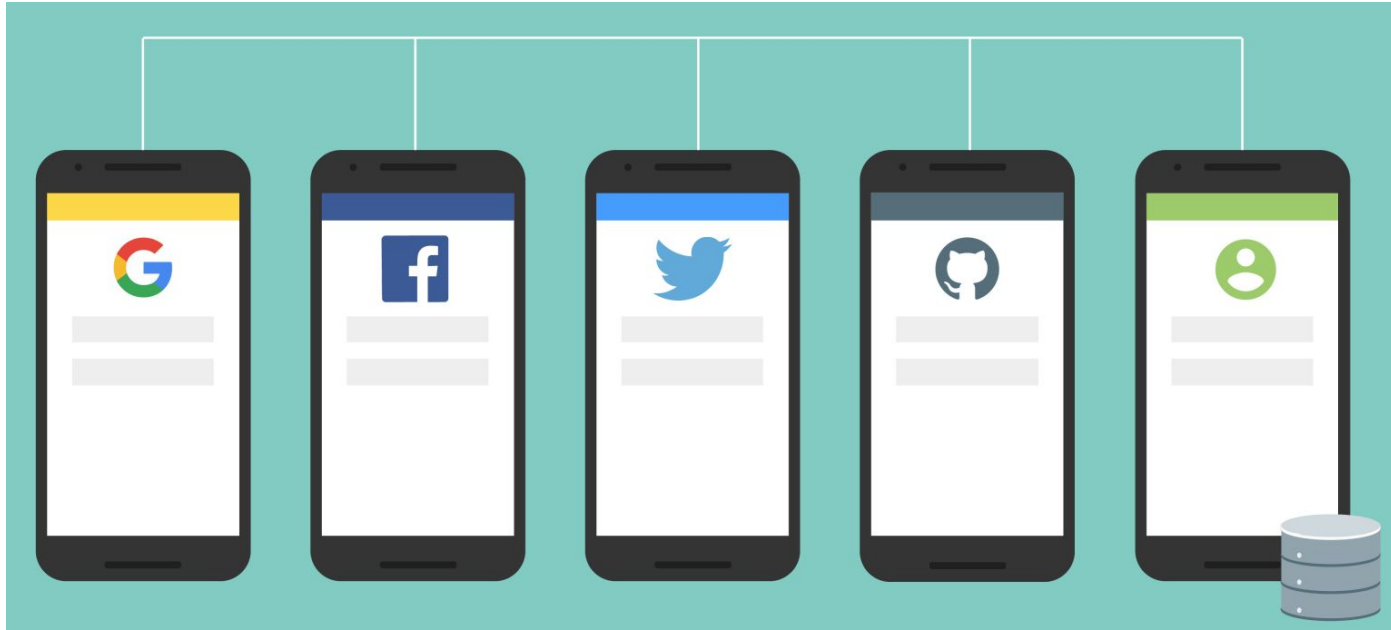
Test Lab









Crash Reporting



Authentication



Authentication

Provedor	Status
 E-mail/senha	Ativado
 Google	Ativado
 Facebook	Ativado
 Twitter	Ativado
 GitHub	Ativado
 Anônimo	Ativado

- Facebook, Twitter, Github e Google Sign-In;
- Verificação de e-mail, contas anônimas e vínculos de contas;
- Criação de contas manualmente.



Authentication

```
authListener = new FirebaseAuth.AuthStateListener() {  
    @Override  
    public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {  
        FirebaseUser user = firebaseAuth.getCurrentUser();  
        if (user != null) {  
            //Logged in;  
        } else {  
            //Logged out;  
        }  
    }  
};
```

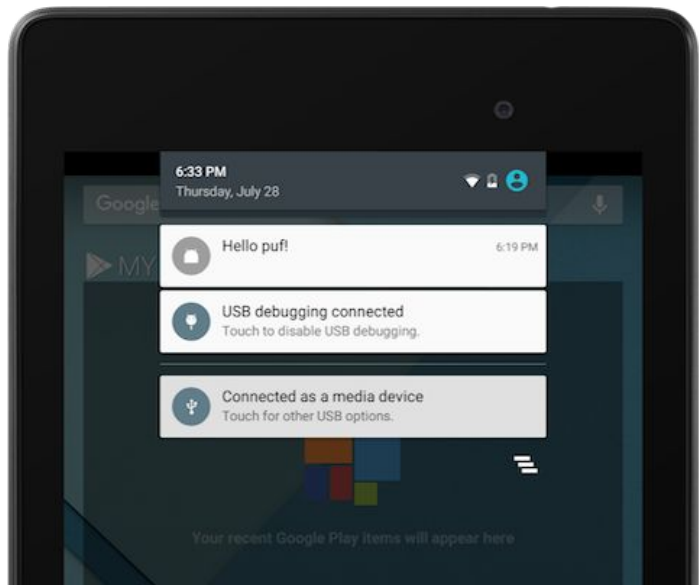


Authentication

```
FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();  
if (user != null) {  
    String name = user.getDisplayName();  
    String email = user.getEmail();  
    Uri photoUrl = user.getPhotoUrl();  
}
```



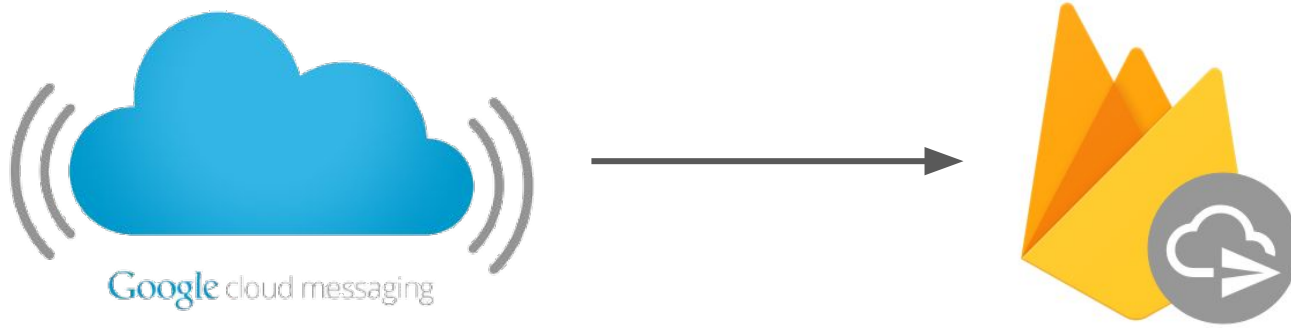
Firebase Cloud Messaging (FCM)



- Envio de mensagens e notificações por segmentação, ou dispositivo único;
- Integração automática com Analytics;
- Fila, tempo de expiração, verificações etc.



Migrando



developers.google.com/cloud-messaging/android/android-migrate-fcm



Realtime Database



- Sincronizado com multiplos devices;
- Resolução de conflitos;
- Adeus "pull to refresh".

Realtime Database

```
FirebaseDatabase database = FirebaseDatabase.getInstance();  
DatabaseReference myRef = database.getReference("palestras");  
  
myRef.setValue("Oi, Android Dev Conference!");
```

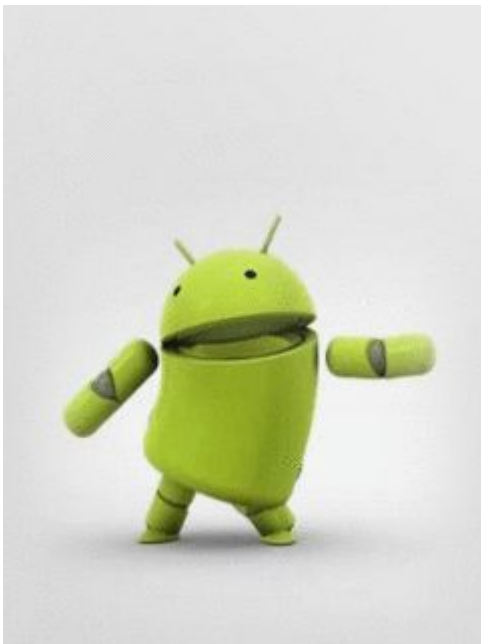


Realtime Database

```
myRef.addValueEventListener(new ValueEventListener() {  
    @Override  
    public void onDataChange(DataSnapshot dataSnapshot) {  
        String value = dataSnapshot.getValue(String.class);  
        Log.d(TAG, "Valor: " + value);  
    }  
  
    @Override  
    public void onCancelled(DatabaseError error) {  
        Log.w(TAG, "Deu ruim.", error.toException());  
    }  
});
```



E se a conexão morrer?

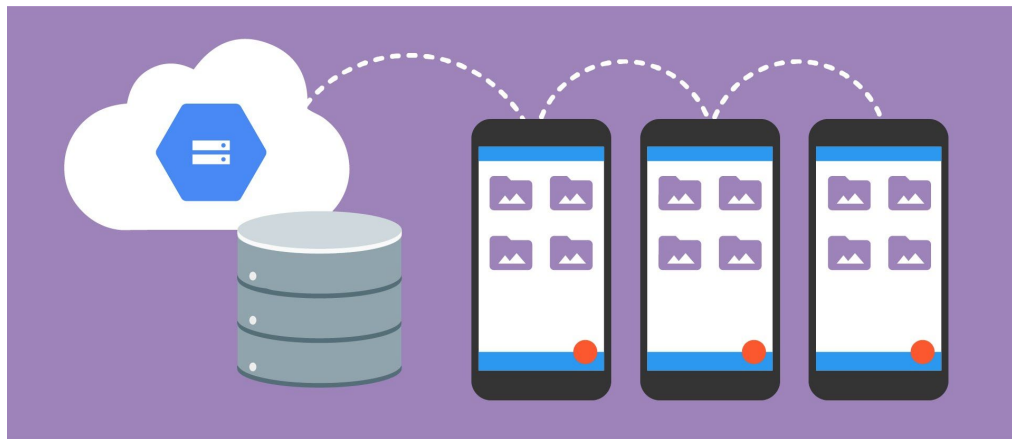


O Firebase Realtime irá funcionar mesmo offline, utilizando o cache de informações.

Quando a rede do aparelho voltar, os dados serão automaticamente sincronizados.



Storage



- Audio, video e imagem;
- Integrado com *Authentication*;
- Escalável;
- *Upload e download* robusto.



Storage

```
FirebaseStorage storage = FirebaseStorage.getInstance();

StorageReference storageRef = storage.getReferenceFromUrl("gs://<your-bucket-name>");

//Criando uma nova referência.
StorageReference imagesRef = storageRef.child("images");

//Com paths!
StorageReference spaceRef = storageRef.child("images/space.jpg");
```



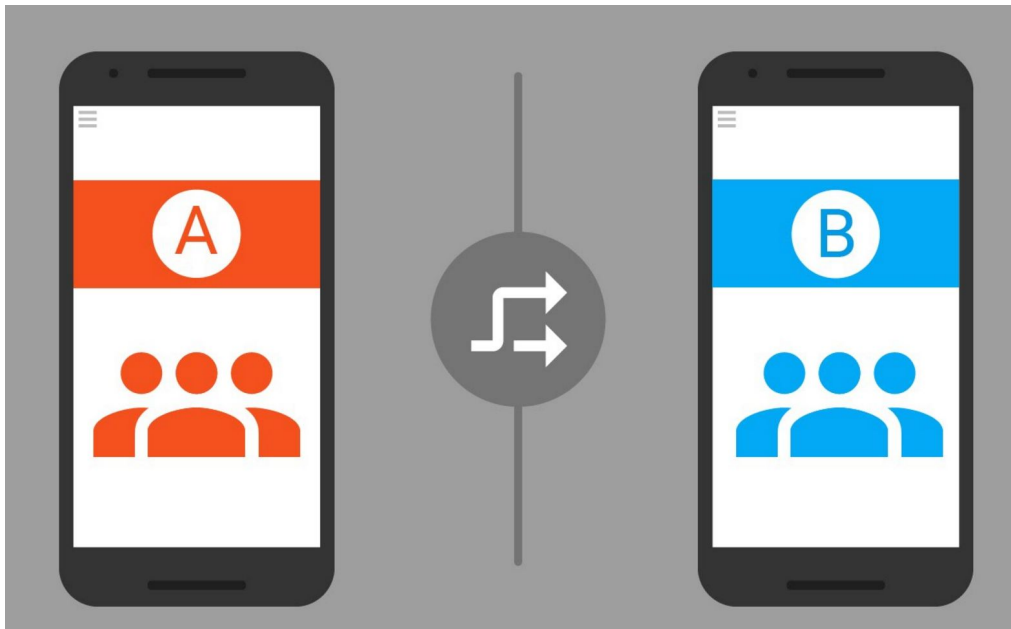
Storage

```
Uri file = Uri.fromFile(new File("path/to/images/dev_conference.jpg"));
StorageReference riversRef = storageRef.child("images/" + file.getLastPathSegment());

uploadTask = riversRef.putFile(file);
uploadTask.addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception exception) {...})
.addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
    @Override
    public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
        Uri downloadUrl = taskSnapshot.getDownloadUrl();
    }
});
```



Remote Config

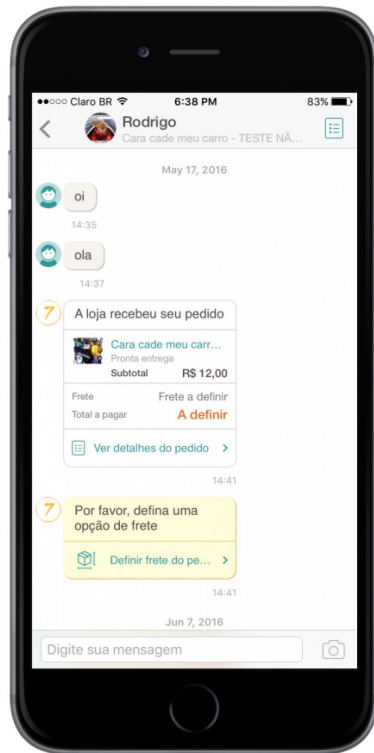


- Não há necessidade de atualização do app;
- Tempo de expiração;
- Teste a/b;
- Validar com poucos usuários;
- Ligar e desligar features.

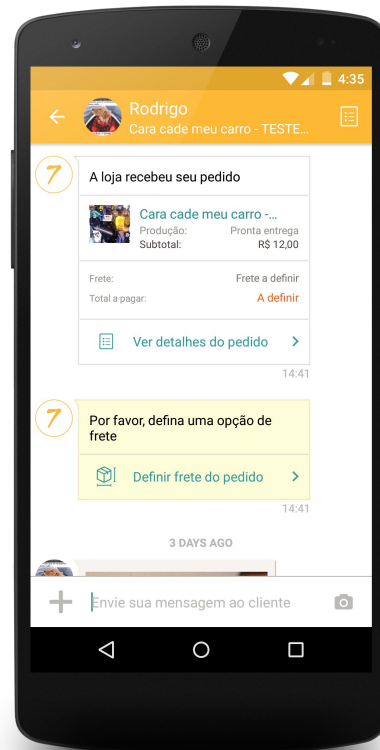


Remote Config

Desligado



Ligado



Remote Config

```
firebaseRemoteConfig = FirebaseRemoteConfig.getInstance();  
  
firebaseRemoteConfig.getString(MY_KEY);  
firebaseRemoteConfig.getBoolean(OTHER_KEY);  
...
```



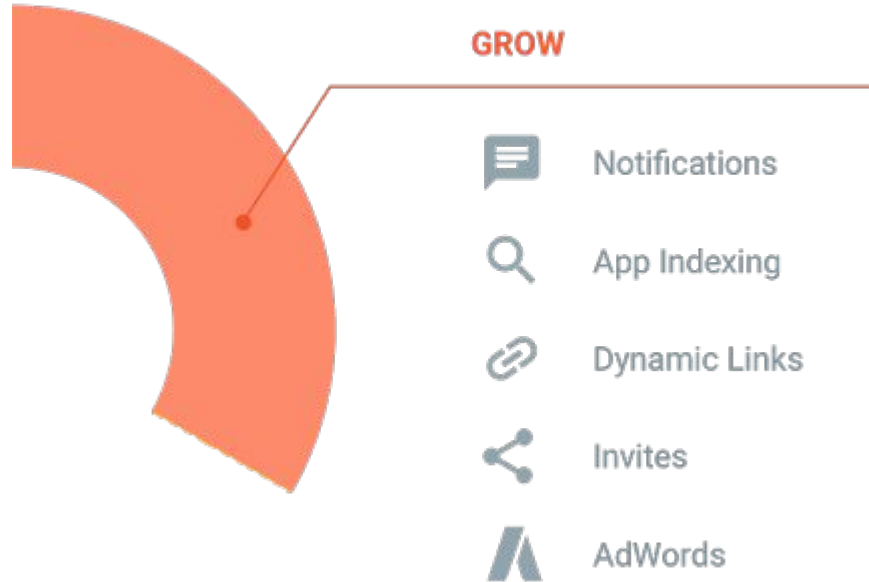
Test Lab



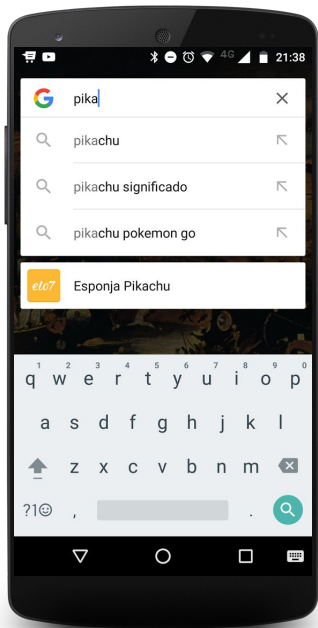
- Relatórios;
- *Android Studio*;
- Custom scripts.



Give me users, please



App Indexing



- Reengajando os usuários;
- Integrado com o Google Search;
- Indexa o conteúdo da aplicação;
- Aumenta o ranking nas buscas.



App Indexing

```
public class MainActivity extends Activity {
    private GoogleApiClient client;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
        client = new GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
    }

    public Action getAction() {
        Thing object = new Thing.Builder()
            .setName("Titulo")
            .setDescription("Alguma descricao")
            .setUrl("androiddevbr.slack.com")
            .build();

        return new Action.Builder(Action.TYPE_VIEW)
            .setObject(object)
            .setActionStatus(Action.STATUS_TYPE_COMPLETED)
            .build();
    }
}
```



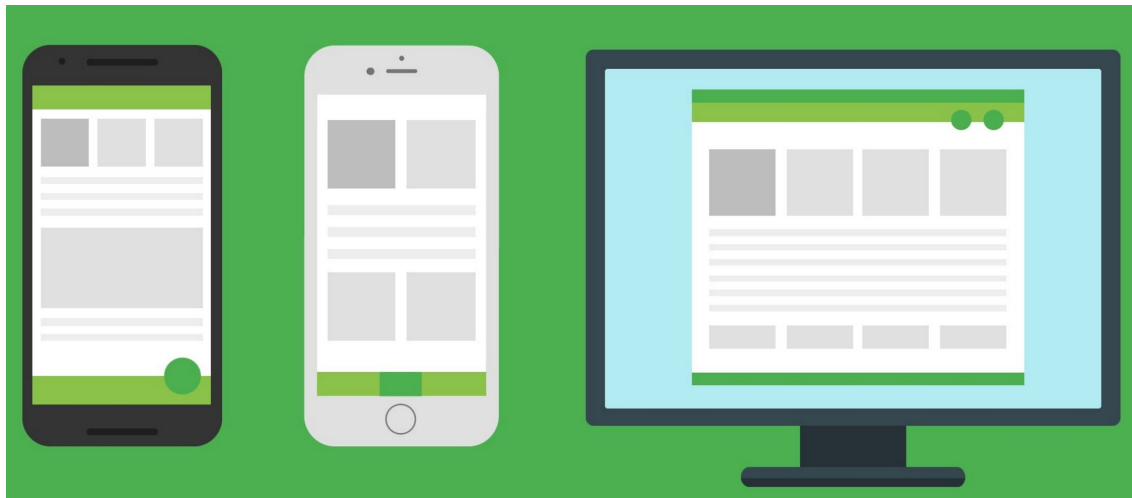
App Indexing

```
@Override
public void onStart() {
    super.onStart();
    client.connect();
    AppIndex.AppIndexApi.start(client, getAction());
}
```

```
@Override
public void onStop() {
    AppIndex.AppIndexApi.end(client, getAction());
    client.disconnect();
    super.onStop();
}
```



Dynamic Links



Dynamic Links são URL's que podem, dinamicamente, alterar o comportamento para fornecer a melhor experiência em diferentes plataformas.



Dynamic Links

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    ...

    // Inicia o Google API Client com o AppInvite API;
    client = new GoogleApiClient.Builder(this)
        .enableAutoManage(this, this)
        .addApi(AppInvite.API)
        .build();
}
```

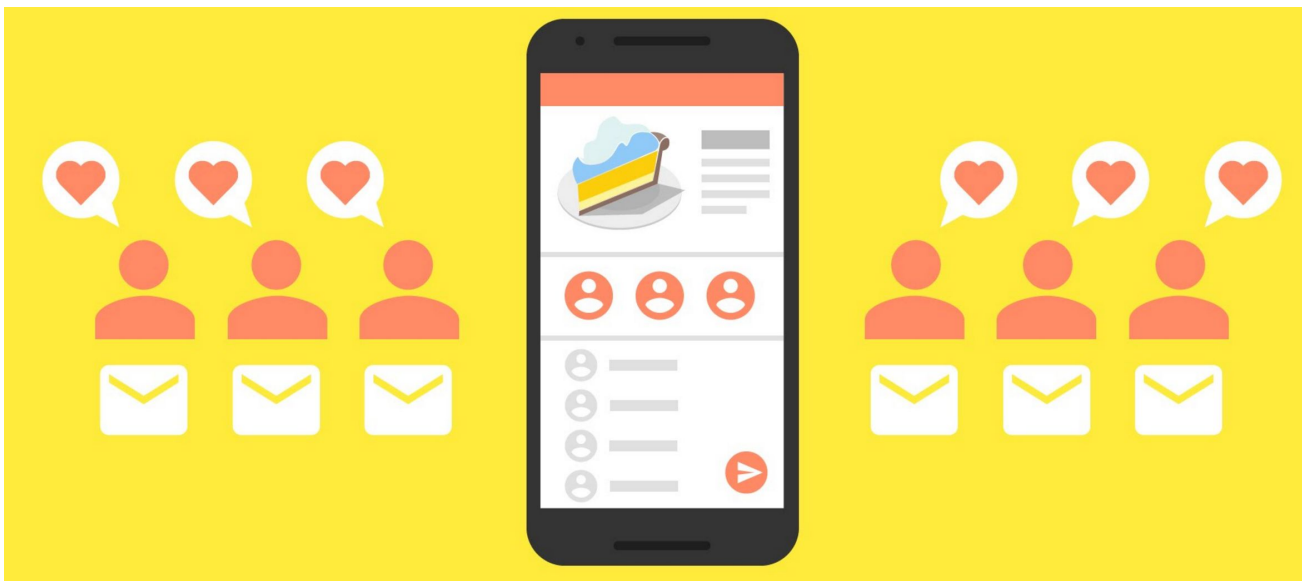


Dynamic Links

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    // Verificando se o app foi iniciado via deeplink;
    boolean autoLaunchDeepLink = false;
    AppInvite.AppInviteApi.getInvitation(client, this, autoLaunchDeepLink)
        .setResultCallback(
            new ResultCallback<AppInviteInvitationResult>() {
                @Override
                public void onResult(@NonNull AppInviteInvitationResult result) {
                    if (result.getStatus().isSuccess()) {
                        Intent intent = result.getInvitationIntent();
                        String deepLink = AppInviteReferral.getDeepLink(intent);
                    } else {
                        // ...
                    }
                }
            }
        );
}
```



Invites



Invites

```
private void onInviteClicked() {  
    Intent intent =  
        new AppInviteInvitation.IntentBuilder(getString(R.string.invitation_title))  
            .setMessage(getString(R.string.invitation_message))  
            .setDeepLink(Uri.parse(getString(R.string.invitation_deep_link)))  
            .setCustomImage(Uri.parse(getString(R.string.invitation_custom_image)))  
            .setCallToActionText(getString(R.string.invitation_cta))  
            .build();  
    startActivityForResult(intent, REQUEST_INVITE);  
}
```



Invites

```
...
@Override
public void onResult(AppInviteInvitationResult result) {
    if (result.getStatus().isSuccess()) {
        // Extraíndo a Intent
        Intent intent = result.getInvitationIntent();
        String invitationId = AppInviteReferral.getInvitationId(intent);
    }
}
```



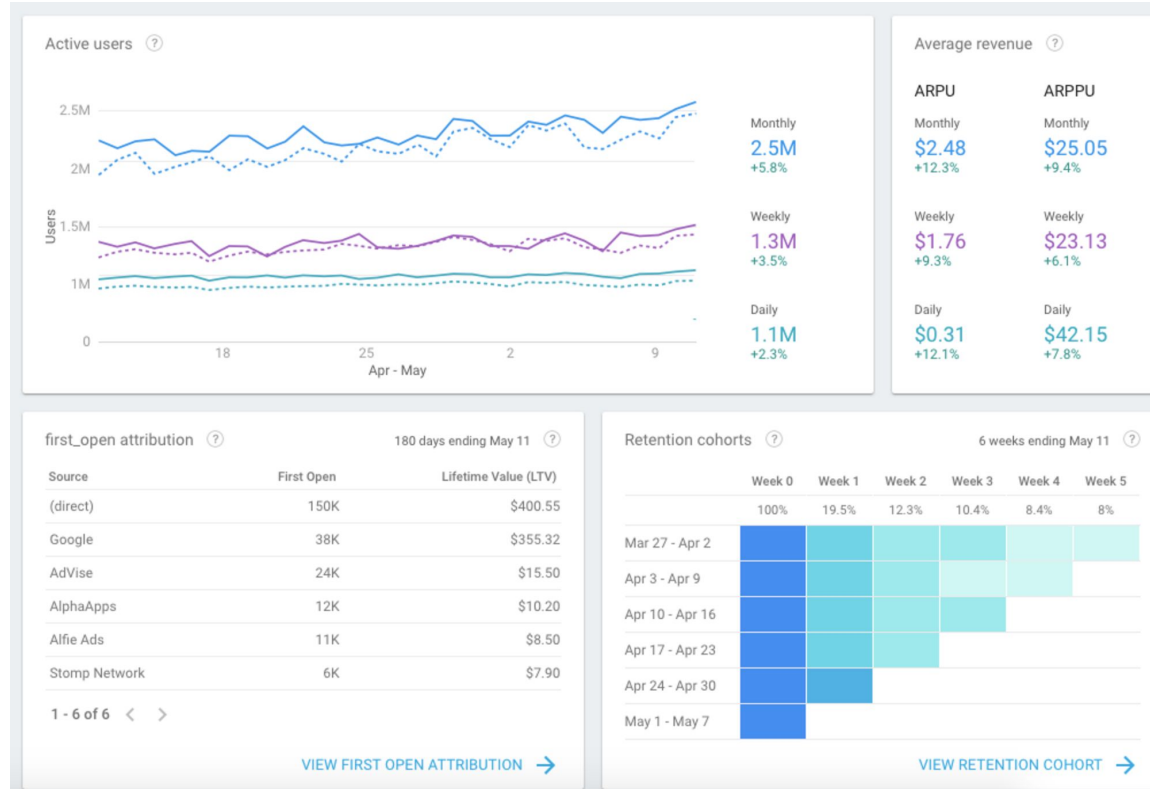
AdWords



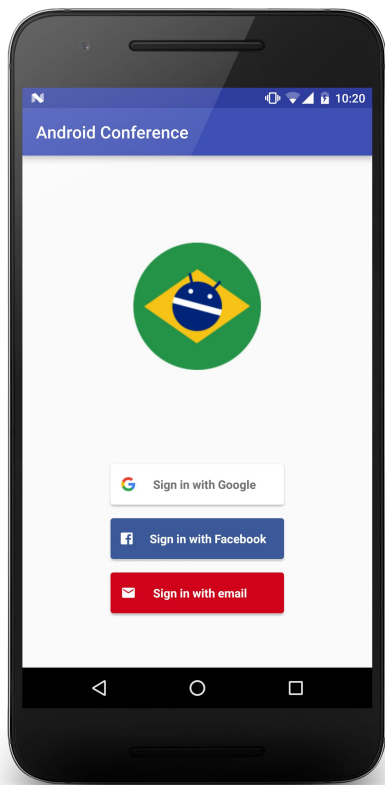
- Exibe anúncios de anunciantes do *Google*;
- UI customizada;



Analytics



Dica!



github.com/firebase/FirebaseUI-Android



Dica!

```
startActivityForResult(  
    AuthUI.getInstance().createSignInIntentBuilder()  
        .setTheme(getSelectedTheme())  
        .setLogo(getSelectedLogo())  
        .setProviders(getSelectedProviders())  
        .setTosUrl(getSelectedTosUrl())  
        .build(),  
    RC_SIGN_IN);
```



É de graça!

Considerações finais

- Explore o Analytics;
- Faça um brainstorm com seu time para entender a real necessidade;
- Comece a validar suas ideias;
- Melhor amigo do freela;
- Divulgue para sua empresa;
- Compartilhe seus problemas com a comunidade!

#firebase

Obrigado!



Rodrigo Sicarelli

br.linkedin.com/in/rodrigodicarelli

[@rsicarelli](https://twitter.com/rsicarelli)

github.com/rsicarelli



Isabela Terribili

br.linkedin.com/in/isabela-cristina-terribili-baro

[@isabelaterribili](https://twitter.com/isabelaterribili)

github.com/isabelaterribili

#AndroidDevConference
Dúvidas?