2. Advanced Install and In-App Event Attribution

Task: Implement install and in-app event attribution into AppsFlyer using the React Native framework.

- **Resources:**
  - React Native Guide
  - Sample Fruit App
- **Deliverable:**
  - Explain the process of tracking both installs and in-app events with AppsFlyer's SDK integration.
  - Discuss the significance of AppsFlyer's comprehensive attribution model in understanding user behavior and optimizing marketing efforts.
  - **Challenge Addition:** Implement custom in-app event tracking that aligns with specific business goals (e.g., purchase funnel, user retention metrics). Present a report on the insights gained from this data.

Introdução

Este documento tem como objetivo passar uma visão geral sobre a implementação do AppsFlyer em aplicativos de React Native.
Com a ferramenta será possível acompanhar a performance de campanhas voltadas para o aplicativo, trazendo a visão sobre como a jornada do usuário é impactada por diferentes canais e estratégias.

Requisitos mínimos

O processo de implementação da biblioteca exige a vinculação de uma conta do AppsFlyer com o aplicativo.

Cadastramento do App

Para isso é necessário realizar o cadastramento do aplicativo para cada plataforma alvo (Android e iOS) conforme especificado pela documentação abaixo.
[Adding an app to AppsFlyer]

Dev Key

Uma vez cadastrado, é necessário obter a **dev key** que será utilizada posteriormente na integração da SDK
[Basic SDK integration guide - Retrieve the dev key]

Instalando o SDK

Após isso é necessário realizar a instalação da biblioteca no projeto do app conforme especificado na documentação abaixo:
Adding react-native-appsflyer to your project
Os desenvolvedores devem determinar a melhor maneira de implementar, utilizando autolink ou de forma manual. A versão da biblioteca também deve ser levada em consideração

Configuração de Inicialização

Após instalada, deve ser adicionado o código para inicialização da biblioteca no aplicativo. A recomendação é que este código seja executado assim que o app for iniciado para garantir que os eventos sejam devidamente capturados.
Após instalação realizada, os eventos de instalação serão automaticamente capturados com suas respectivas origens.
Abaixo o link para documentação da inicialização da SDK é um exemplo do código a ser executado:
Basic integration of the SDK

```
JavaScript
import appsFlyer from 'react-native-appsflyer';

appsFlyer.initSdk(
  {
    devKey: 'K2***********99',
    isDebug: false,
    appId: '41*****44',
    onInstallConversionDataListener: true, //Optional
    onDeepLinkListener: true, //Optional
    timeToWaitForATTUserAuthorization: 10 //for iOS 14.5
  },
  (result) => {
    console.log(result);
  },
  (error) => {
    console.error(error);
  }
);
```

Os parâmetros a serem preenchidos são:

- **devKey:** Referente a chave extraída no passo anterior
- **appId:** Para dispositivos iOS, é o mesmo id utilizado para cadastramento do app na ferramenta
- **isDebug:** Recomendamos configurar o valor para mudar dinamicamente conforme o ambiente do app, evitando o risco de subir o app em debug no ambiente produtivo.
- **onInstallConversionDataListener:** Caso haja necessidade para utilizar a conversion data no contexto do app.
- **onDeepLinkListener:** Caso o app tenha a funcionalidade de deep link.
- **timeToWaitForATTUserAuthorization:** Caso o IDFA possa ser utilizado, adicionamos a quantidade de tempo para espera da popup de autorização.

Validação

Após a implementação podemos validar se está tudo funcionando através de dispositivos de teste. Abaixo links de referência para documentação sobre como cadastrar dispositivos de homologação e realizar testes da SDK:

Registering test devices

Testing the SDK integration for marketers

In-App Events

Os In-App eventos são utilizados para coletar dados sobre as interações do usuário com o aplicativo (cliques em botão, adição de produtos ao carrinho, compras, cadastros). Estas informações podem ser repassadas via postback para os veículos de mídia, permitindo enriquecimento dos dados de campanha.

Mapeamento dos eventos a serem coletados

Antes de tudo, é necessário listar quais interações são relevantes para captura no aplicativo. Durante esta etapa geralmente recomendamos construir um documento listando os eventos e propriedades que desejamos coletar (exemplo abaixo).

event_name	params	type	example value
af_click_on_item	name	string	Banana
	image	string	https://cdn.mos.cms.futurecdn.net/42E9as7NaTaAi4A6JcuFwG-1200-80.jpg
	price	float	10.0
	info	string	Go bananas!
af_added_to_cart	name	string	Banana
	image	string	https://cdn.mos.cms.futurecdn.net/42E9as7NaTaAi4A6JcuFwG-1200-80.jpg
	price	float	10.0
	info	string	Go bananas!
af_removed_from_cart	name	string	Banana
	image	string	https://cdn.mos.cms.futurecdn.net/42E9as7NaTaAi4A6JcuFwG-1200-80.jpg
	price	float	10.0
	info	string	Go bananas!
af_view_cart	cart_size	int	3

af_check_out	product_list	list<product>	[
			{
			name: 'Banana',
			image:
			'https://cdn.mos.cms.futurecdn.net/42E9as7NaTaAi4A6JcuFwG-1200-80.jpg',
			price: 10.0,
			info: 'Go bananas!',
			},
			{
			name: 'Strawberry',
			image: 'https://images.unsplash.com/photo-1467825487722-2a7c4cd62e75',
			price: 11.0,
			info: 'Strawberry Fields Forever!',
			}
			]

	af_revenue	float	21.0

A referência de estrutura de eventos predefinidos pode auxiliar neste mapeamento:
[In-app events—Event structure]

Implementação

Após mapeados, os eventos devem ser implementados manualmente pelos desenvolvedores nas etapas onde as interações com o aplicativo ocorrem.
O disparo de cada evento é feito pela função `logEvent`. Abaixo um exemplo de código demonstrando um envio:

```javascript
appsFlyer.logEvent(
  "af_added_to_cart",
  {
      name: 'Banana',
      image:
'https://cdn.mos.cms.futurecdn.net/42E9as7NaTaAi4A6JcuFwG-1200-80.jpg',
      price: 10,
      info: 'Go bananas!',
  }
);
```

Configuração de postback na interface

Uma vez implementados, é possível configurar o postback dos eventos para as ferramentas de mídia integradas.
Dentre as opções de configuração é possível definir a janela de envio, quais eventos devem ser enviados, se o envio deve ocorrer em todos ou apenas para eventos atribuídos ao parceiro.
Outra configuração importante é o mapeamento de parâmetros dos eventos. Cada mídia possui sua própria para os nomes de eventos e parâmetros aceitos, então precisamos "traduzir" o que é coletado pelo AppsFlyer para que possam ser lidos corretamente.
Abaixo o link da documentação oficial com maiores detalhes sobre o postback de eventos.
In-app event postback configuration