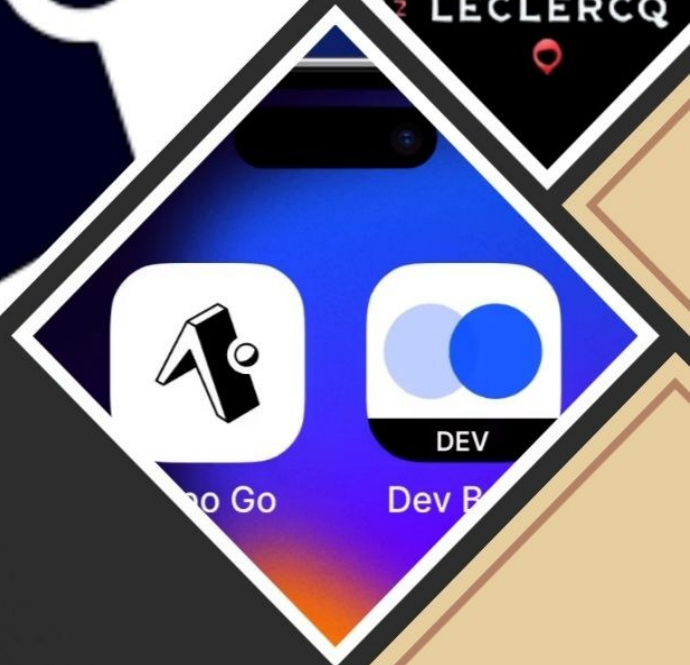




FEDNAI
LECLERCQ



VEILLE TECHNOLO GIQUE

Presented by :

fednail leclercq

Table des matières

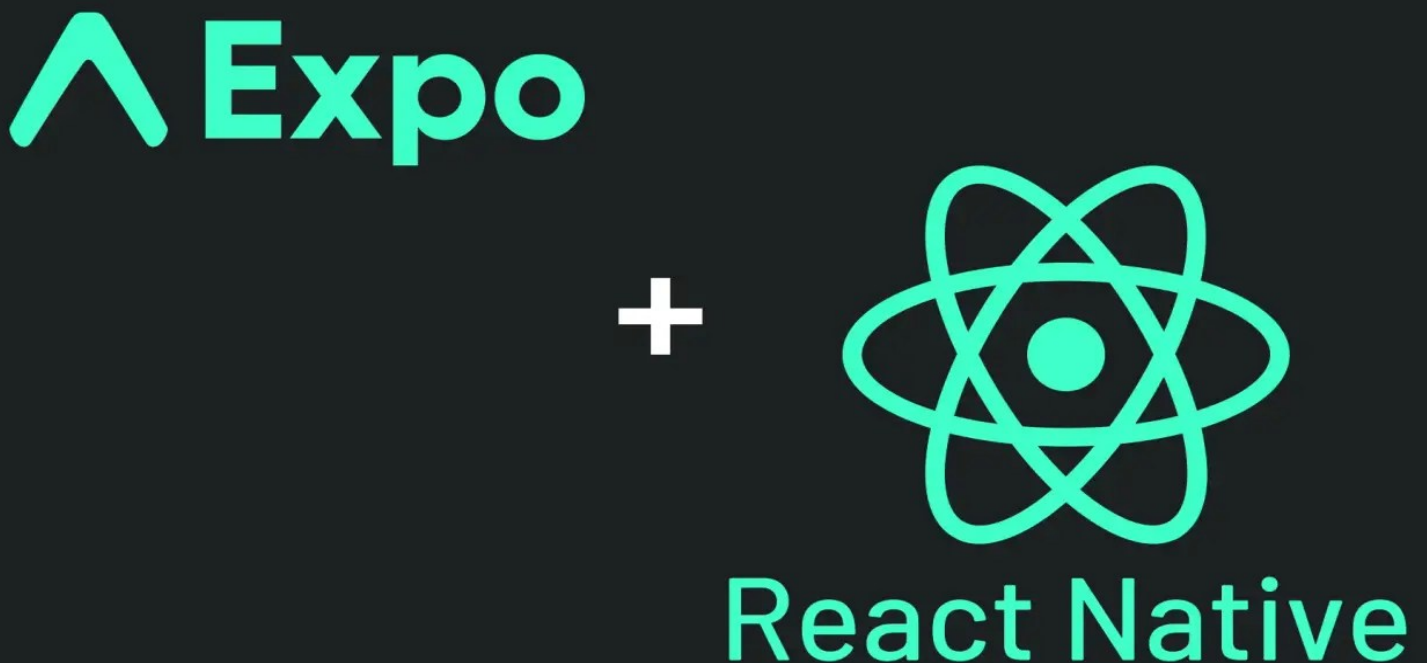
Introduction	2
Fonctionnement	3
Principe général	3
Architecture simplifiée.....	3
Managed Workflow et Bare Workflow	3
Rôle d'Expo Go	5
Outils principaux.....	5
Avantages 👍	6
Inconvénients 🚫	6
Cas d'usage	6
Exemple d'application faite avec React Native et Expo.....	7
Comparaison	9
Évolutions récentes	9
Évolutions récentes du SDK Expo (2023 – 2026).	10
Interprétation simple	10
Tendance actuelle :	10
Exemple simple de code Expo	11
Conclusion.....	11
Recommandation	12
Sources	12

Veille technologique : Expo et Expo Go dans l'écosystème React Native

Expo est un Framework open source qui simplifie la création d'applications mobiles avec React Native, en réduisant fortement la configuration native au démarrage. Expo Go est l'application compagnon qui permet de tester rapidement un projet Expo sur un téléphone ou un émulateur, sans compiler toute l'application à chaque modification.

Introduction

Expo existe pour accélérer le développement mobile et rendre React Native plus accessible, surtout pour les prototypes, les projets étudiants et les équipes qui veulent aller vite. L'idée principale est de laisser le développeur se concentrer sur le JavaScript/TypeScript, tandis qu'Expo gère une grande partie des détails techniques liés à iOS et Android.



Expo Go est une application gratuite et open source, disponible sur les stores, qui sert de “bac à sable” pour exécuter et tester un projet Expo sur appareil réel. Elle est utile pour voir immédiatement les changements, mais elle ne permet pas d'utiliser n'importe quelle librairie native personnalisée ni de modifier directement le code natif.

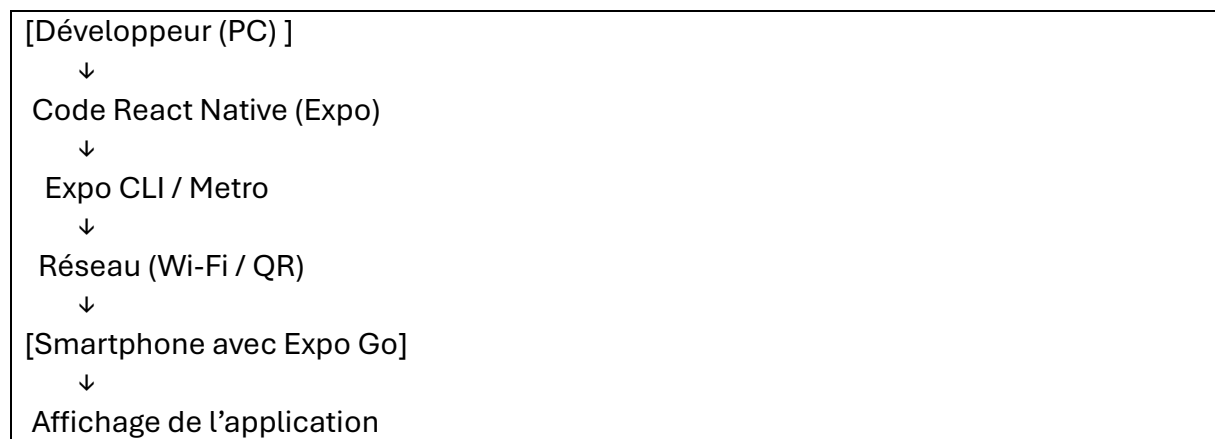
Fonctionnement

Principe général

Dans un projet Expo, on écrit l'interface et la logique applicative en React Native, puis Expo fournit un ensemble d'outils, de bibliothèques et de services pour lancer, tester et générer l'application. La documentation officielle décrit aussi un workflow moderne basé sur les projets générés à la demande avec `npx expo prebuild`, les `development builds` et `EAS Build` pour les binaires.

Architecture simplifiée

On peut résumer Expo comme une couche au-dessus de React Native :

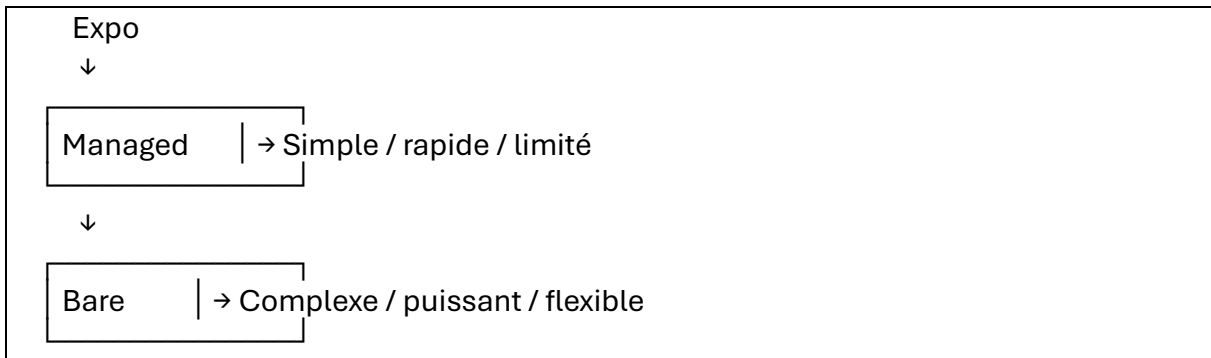


Expo fournit des modules prêts à l'emploi pour accéder à des fonctions du téléphone comme la caméra, le stockage local ou la géolocalisation, ce qui évite souvent d'écrire du code natif soi-même.

Managed Workflow et Bare Workflow

Le **Managed Workflow** est le mode le plus simple : Expo prend en charge une grande partie de la configuration native, et le projet se gère surtout côté JavaScript/TypeScript. C'est le mode recommandé pour démarrer vite et éviter la complexité de Xcode et Android Studio au quotidien.

Le **Bare Workflow** correspond à un projet où les dossiers natifs android et ios sont versionnés et maintenus manuellement. Ce mode donne plus de liberté, mais demande plus de maintenance et convient mieux aux projets qui ont besoin d'intégrations natives avancées.



Critère	Managed Workflow	Bare Workflow
Code natif visible	Non, ou très peu	Oui
Configuration	Simplifiée	Plus technique
Flexibilité	Bonne pour la plupart des apps	Très élevée
Maintenance	Plus faible	Plus forte

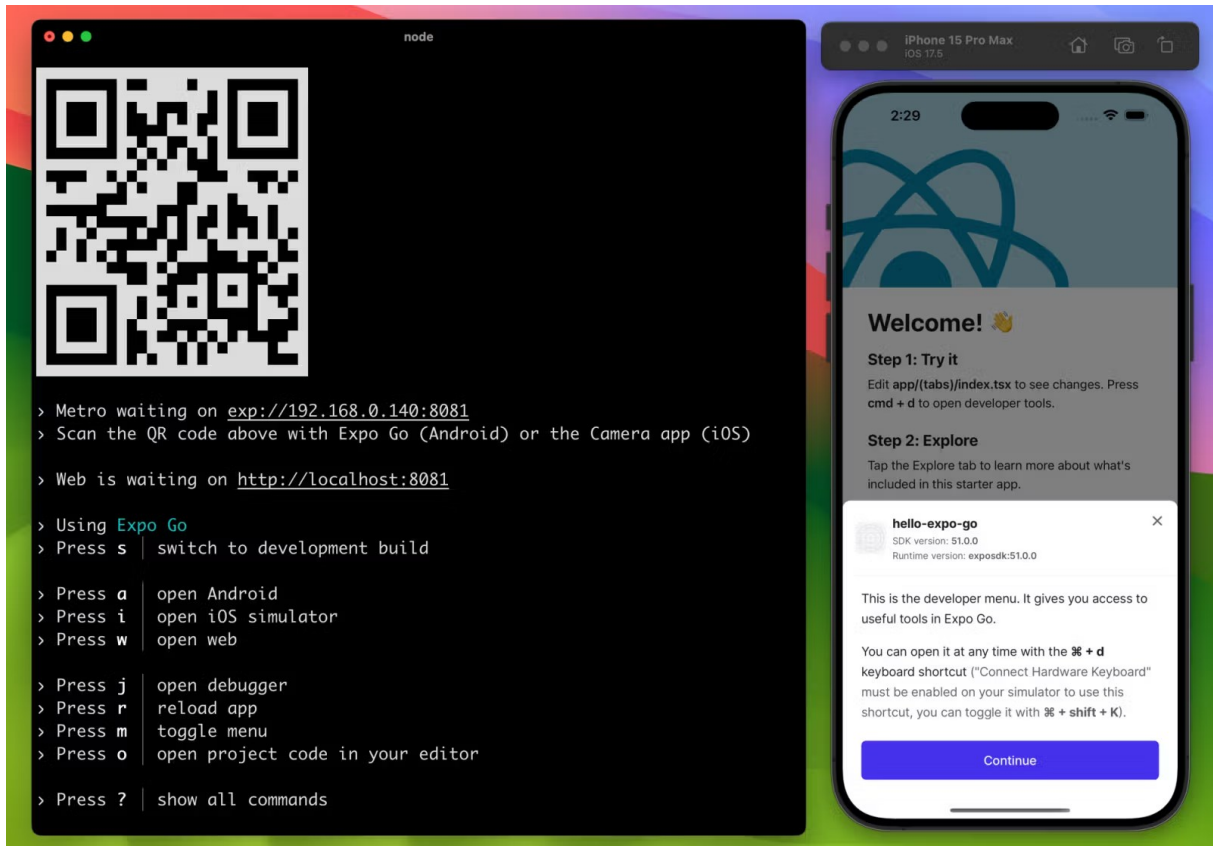
Workflow comparison

Feature	Managed workflow	Bare workflow
Develop apps with only JavaScript/TypeScript	✓	
Use Expo build service to create your iOS and Android builds	✓	✓ (with EAS Build)
Use Expo's push notification service	✓	✓
Use Expo's over the air updates features	✓	✓
Develop with the Expo Go app	✓	✓ (if you follow these guidelines)
Access to Expo SDK	✓	✓
Add custom native code and manage native dependencies		✓
Develop in Xcode and Android Studio		✓

Tableau comparatif Workflow

Rôle d'Expo Go

Expo Go permet de tester un projet sans compilation complexe, ce qui est idéal pour valider rapidement l'interface et la logique. En revanche, la documentation précise qu'Expo Go ne supporte pas les bibliothèques tierces qui nécessitent du code natif personnalisé, et il n'est pas adapté aux applications de production complexes.



(Image du lance expo go sur différence appareil).

Outils principaux

Les outils les plus importants de l'écosystème Expo sont :

- Create-expo-app, pour initialiser un projet.
- La CLI Expo, pour lancer, prévisualiser et préparer l'application.
- Les modules Expo, pour accéder à des fonctionnalités du téléphone.
- EAS Build, pour générer des applications natives dans le cloud.
- Les development builds, qui remplacent souvent Expo Go dès qu'on a besoin de modules natifs personnalisés.

Avantages

Expo est apprécié pour sa simplicité de prise en main et sa rapidité de démarrage. La documentation officielle insiste sur le fait qu'on peut lancer rapidement un projet, le tester sur un appareil et itérer sans configurer toute la chaîne native au départ.

Expo fait aussi gagner du temps sur les tâches répétitives : configuration, génération des binaires, compatibilité de certains modules, et mise à jour de projets. Pour un étudiant BTS SIO, cela permet de se concentrer davantage sur la logique applicative, l'interface et les bonnes pratiques React Native.

Inconvénients

La principale limite d'Expo Go est qu'il n'accepte pas les bibliothèques qui nécessitent un code natif spécifique, donc il ne couvre pas tous les cas d'usage avancés. Pour certaines apps, il faut passer à une development build ou au Bare Workflow.

Expo implique aussi une certaine dépendance à son écosystème : on bénéficie de ses outils, mais on suit aussi son rythme d'évolution et ses contraintes de compatibilité. En pratique, cela reste très bon pour la majorité des projets, mais moins libre qu'un projet React Native CLI entièrement natif.

Cas d'usage

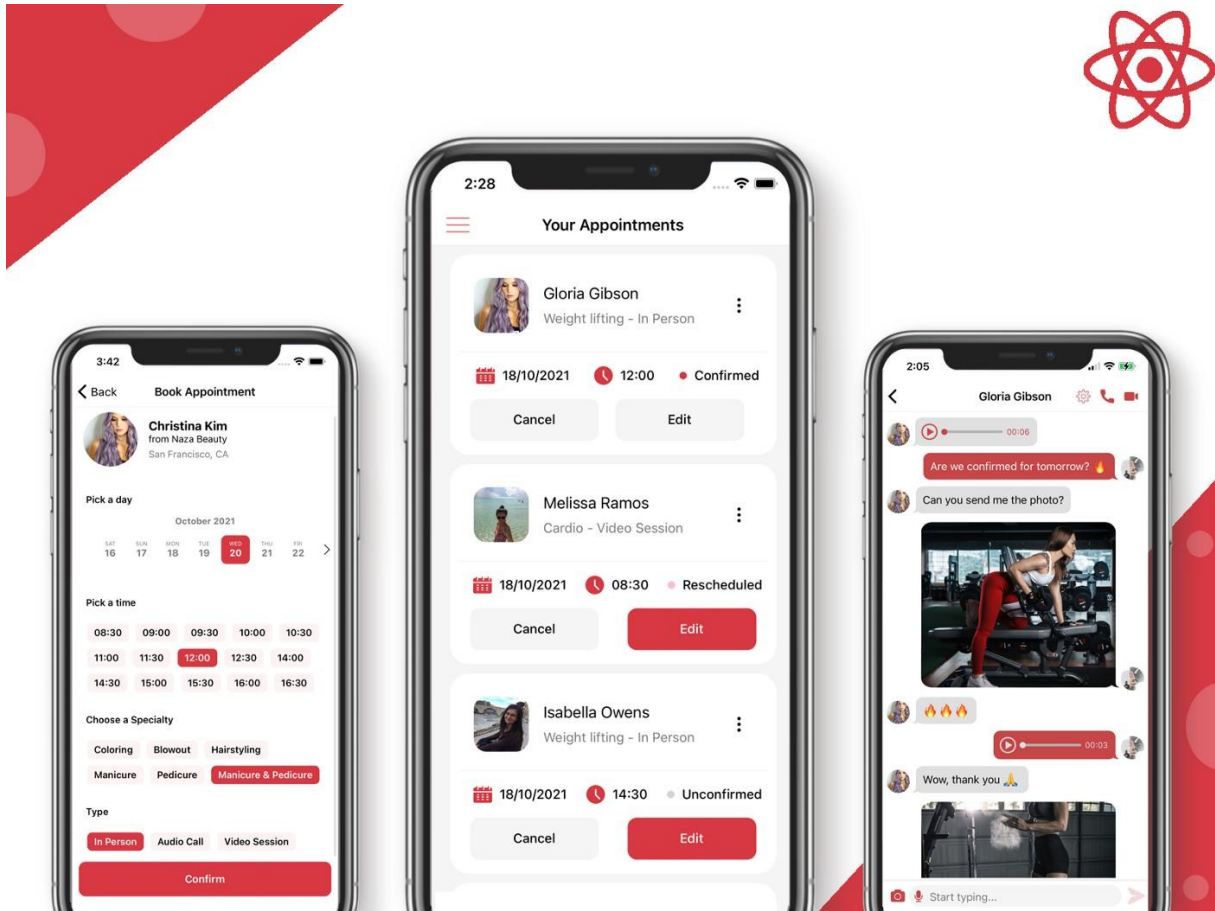
Expo est particulièrement adapté aux projets suivants :

- Application simple ou moyenne complexité ;
- Prototype rapide ;
- Projet étudiant ;
- MVP ;
- Application interne ;
- Projet où la vitesse de développement est prioritaire.

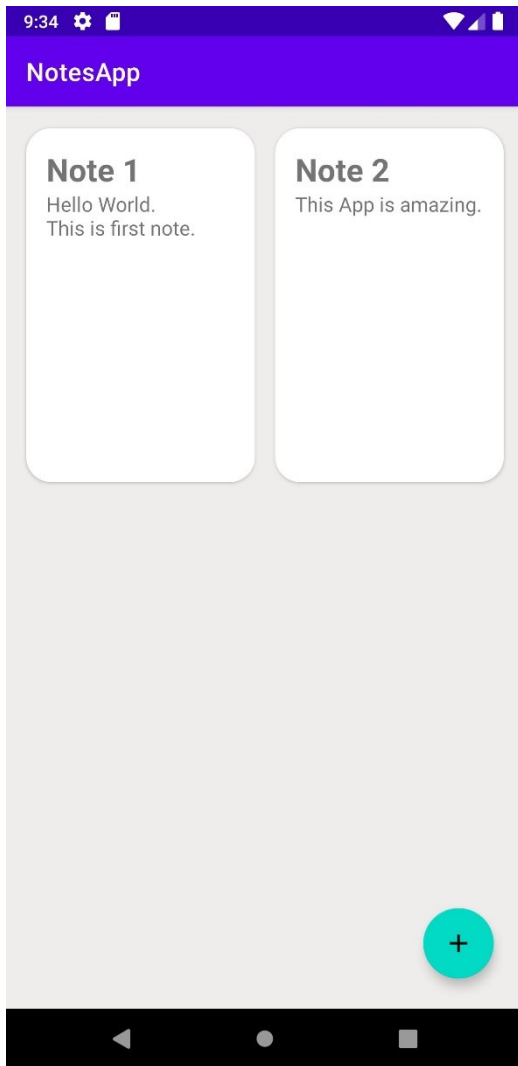
Exemples concrets :

- Une application de gestion de tâches ;
- Un carnet de notes ;
- Une application de réservation simple ;
- Un prototype pour un dossier BTS ;
- Une démo technique à présenter rapidement.

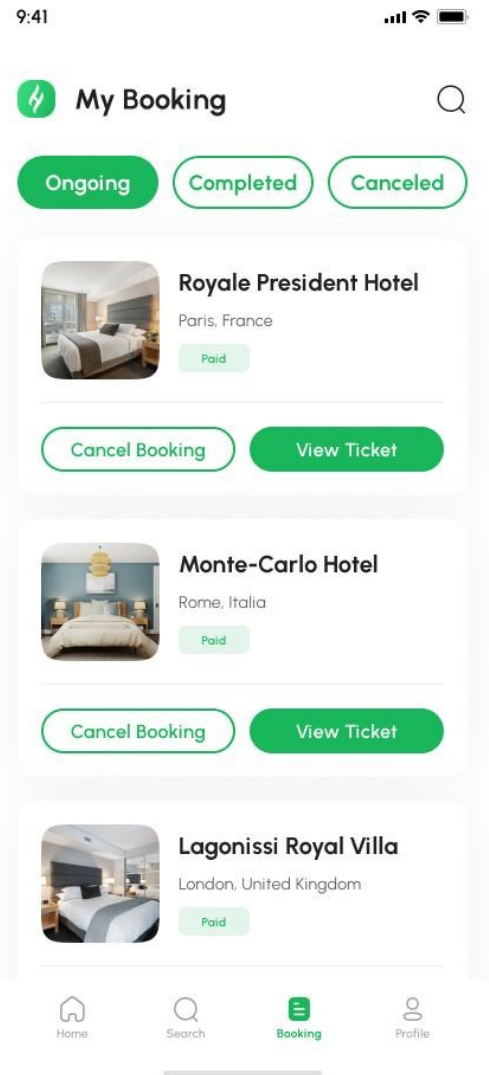
Exemple d'application faite avec React Native et Expo



(Image de reservation de coach sportif avec Expo et React Native).



(Appli de note faite avec Expo & React Native).



(Image de réservation de chambre avec React Native et Expo).

Comparaison

Critère	Expo	React Native CLI	Flutter
Facilité d'utilisation	Très élevée	Moyenne	Élevée
Performance	Bonne pour la plupart des apps	Bonne à très bonne	Très bonne
Flexibilité	Bonne, mais encadrée	Très élevée	Élevée
Courbe d'apprentissage	Faible à moyenne	Plus élevée	Moyenne

Expo est le plus simple pour démarrer rapidement dans l'univers React Native, alors que React Native CLI donne plus de contrôle sur les projets natifs. Flutter est une alternative différente, avec son propre langage et son propre moteur de rendu, ce qui peut être un avantage pour certaines équipes, mais demande d'adopter un écosystème distinct.

Évolutions récentes

La documentation d'Expo publiée en 2026 met en avant un workflow moderne centré sur les development builds, EAS Build et expo prebuild, avec Expo Go davantage positionné comme outil d'apprentissage et de test rapide que comme environnement de production.



Les versions récentes montrent aussi que l'écosystème évolue vite : Expo SDK 55 est sorti en février 2026, avec React Native 0.83 et React 19.2, et **l'équipe annonce déjà des versions futures comme SDK 56 au deuxième trimestre 2026**. La page de création de projet indique aussi une transition entre SDK 54 et SDK 55, avec des consignes spécifiques selon l'usage d'Expo Go, ce qui montre une évolution active et structurée du framework

Évolutions récentes du SDK Expo (2023 – 2026).

2023

- Expo SDK 48 — 24 févr. 2023
- Expo SDK 49 — 05 juil. 2023

2024

- Expo SDK 50 — 18 janv. 2024
- Expo SDK 51 — 07 mai 2024
- Expo SDK 52 — 12 nov. 2024

2025

- Expo SDK 53 — 30 avr. 2025
- Expo SDK 54 — 10 sept. 2025

2026

- Expo SDK 55 — 25 fév. 2026

Interprétation simple

- Depuis 2023, Expo publie **environ trois versions majeures par an**, chacune apportant des améliorations, compatibilités ou évolutions avec React Native et des API natives.
- Par exemple, **SDK 50** a introduit de nouveaux outils et une meilleure API 📦, tandis que **SDK 51, 52 et 53** ont continué le renforcement de l'architecture et des possibilités pour les développeurs.

Tendance actuelle :

On observe aussi une tendance nette : Expo Go reste utile, mais les projets sérieux basculent de plus en plus vers les **development builds**, car elles permettent d'utiliser des bibliothèques natives personnalisées tout en gardant le confort de l'écosystème Expo.

Exemple simple de code Expo

Voici un exemple minimal d'écran React Native dans un projet Expo :

```
import {Text, View} from 'react-native';

export default function App () {
  return (
    <View style= {{flex: 1, justifyContent: 'center', alignItems: 'center'}} >
      <Text>Bonjour depuis Expo ! </Text>
    </View>
  );
}
```

Cet exemple montre le principe de base : on utilise React Native normalement, et Expo fournit le cadre qui facilite l'exécution et le test du projet.

Conclusion

Expo est une solution très pertinente pour créer rapidement une application mobile avec React Native, surtout si l'objectif est de simplifier le développement et de limiter la complexité native. Expo Go est excellent pour tester et apprendre, mais il ne couvre pas tous les besoins techniques avancés.

Recommandation

- **Utiliser Expo** si le projet est simple, rapide à prototyper, ou destiné à un contexte pédagogique.
- **Utiliser development builds ou Bare Workflow** si l'application nécessite des modules natifs spécifiques ou une personnalisation avancée.
- **Utiliser React Native CLI** si l'on veut un contrôle natif maximal dès le départ.

Sources

- <https://docs.expo.dev/workflow/overview/>
- <https://docs.expo.dev/faq/>
- <https://docs.expo.dev/develop/development-builds/introduction/>
- <https://docs.expo.dev/more/glossary-of-terms/>
- <https://docs.expo.dev/build/introduction/>
- <https://play.google.com/store/apps/details?id=host.exp.exponent&hl=fr>
- <https://www.kodix.fr/outils/expo-go-developpement-application-mobile>
- <https://docs.expo.dev/bare/upgrade/>
- <https://docs.expo.dev/get-started/create-a-project/>
- <https://expo.dev/changelog/sdk-55>
- <https://expo.dev/blog/upgrading-to-sdk-55>
- <https://w3r.one/fr/blog/mobile/react-native/expo/introduction-expo-faciliter-developpement-react-native>
- https://www.reddit.com/r/reactnative/comments/1ehx51z/what_is_the_difference_between_react_native_cli/
- <https://alexsoyes.com/veille-technologique/>
- <https://blog.openreplay.com/fr/demarrage-rapide-expo-construire-apps-react-native/>
- <https://exponent.fr.aptoide.com/app>
- <https://orangeloops.com/2024/08/expo-vs-react-native-cli-which-is-best-for-your-mobile-app-development/>
- <https://www.studit.fr/actus/expo-un-outil-de-developpement-mobile-cross-platform-a-la-page>
- <https://oneuptime.com/blog/post/2026-01-15-expo-managed-to-bare-workflow/view>
- <https://github.com/expo/expo/blob/main/guides/Expo%20Documentation%20Writing%20Style%20Guide.md>
- <https://dev.to/expo/expo-sdk-37-is-now-available-69g>

- <https://www.wesvance.com/posts/understanding-expo>
- <https://pagepro.co/blog/how-to-eject-from-expo-managed-workflow-to-bare/>
- <https://github.com/rauldeheer/expo-docs>
- <https://www.youtube.com/watch?v=KBlbkjxNbM>
- <https://www.youtube.com/watch?v=EAWIIFuOZHU>
- <https://expo.dev/changelog/sdk-55-beta>
- <https://www.youtube.com/watch?v=HNTbj3jNJDg>
- <https://app.daily.dev/posts/expo-sdk-55-is-here-react-native-0-83-react-19-2-legacy-architecture-dropped-new-architecture-i-bzoqkykne>
- <https://www.youtube.com/watch?v=DLAYDULdk9A>
- https://www.linkedin.com/posts/mazharuldev_expo-reactnative-mobiledevelopment-activity-7433022855451402241-Tf-5
- <https://x.com/natsumican63/status/2014491984141156606>
- <https://expo.dev/changelog/sdk-55>
- <https://github.com/react-native-community/cli>
- <https://github.com/anisurrahman072/React-Native-News-2024/blob/master/Expo-News/Expo-Go-to-EAS-build.md>
- <https://docs.expo.dev/build/introduction/>
- <https://docs.expo.dev/faq/>
- <https://expo.dev/brand>
- <https://docs.expo.dev/guides/store-assets/>
- <https://expo.dev/blog/expo-go-vs-development-builds>
- <https://w3r.one/fr/blog/mobile/react-native/expo/optimiser-performance-application-expo>

