Application Android : les langages appropriés

Choisir un développement mobile Android permet d’élargir son audience. 80 % des smartphones en fonction en sont équipés. Pour [créer une app](https://fr.goodbarber.com/creer-app/0-creer-une-app) Android qui mobilise l’appareil photo et d’autres fonctionnalités du smartphone, il faut coder en natif. Les langages de programmation les plus communs sont Java, Kotlin, C et C++.  
  
Java est le langage natif officiel d’Android, c’est aussi le plus populaire. Il est supporté par l’environnement de développement Android Studio. Toutefois, c’est un outil de développement assez difficile à maîtriser.  
  
Kotlin a été récemment introduit en tant que langage « officiel » secondaire par Google. Il ressemble à Java à bien des égards, mais il est plus facile d’usage. Ce langage peut-être combiné avec Java. Il s’impose de plus en plus comme le nouveau standard. Il peut être employé sur plusieurs outils et IDE. Avec lui, il faut moins de lignes de code pour obtenir les mêmes fonctionnalités.  
  
C/C++ sont des langages compatibles avec le développement logiciel Android Studio. Il est utilisable avec le NDK Android. Cela permet d’implémenter du code natif avec du C/C++. Le langage C est souvent à la base des jeux 3D. En effet, il pousse au maximum l’optimisation des potentialités des smartphones Android.  
  
C# est un autre langage utilisable. Il est plus simple d’utilisation pour les débutants que C/C++. Il offre également la possibilité de créer des applications mobiles.

L’application web

La Progressive Web App (PWA) s’instaure désormais comme le futur de l’appli web. Le développeur utilise les technologies web classiques pour la concevoir. L’HTML (Hypertext Markup Language) est l’un des outils essentiels. C’est un langage de balisage conçu pour construire et représenter des pages web. Le CSS (Cascading Style Sheet) fait partie du développement logiciel web. Il spécifie la manière dont les éléments HTML d’une page doivent être présentés. Il permet de choisir la police, les caractères, la couleur de fond. Enfin, JavaScript est le langage de programmation web par excellence. Il permet de créer du contenu mis à jour de façon dynamique. Il sert également à contrôler le contenu multimédia, à animer des pages. C’est un dérivé du C, il est aujourd’hui employé en back-end avec les serveurs Node JS.  
  
La web app peut être créée dans un environnement de développement comme Angular ou Ionic. Une PWA, bien que non-native, pourra mobiliser l’appareil photo. Elle est dotée d’autres caractéristiques natives (fonctionnement hors ligne, rapidité).  
  
React native est utilisé pour construire des applis web spécifiques, les apps hybrides. Cet outil de développement offre la possibilité d’aller plus loin dans le natif. Il est possible d’intégrer des composants natifs à l’architecture de l’app. C’est un moyen de pouvoir utiliser les fonctionnalités natives des devices. Il est de plus en plus utilisé pour créer des apps cross-plateformes. Loin devant React native, les stars du moment restent toutefois Cordova et Unity.

Les applications web compatibles Android et iOS

Les applications natives représentent la majorité des applications publiées sur les Stores. Le mobinaute passe 90 % de son temps connecté sur une app. 92 % de ces applications sont des applis natives. [Comment créer une appli smartphone](https://fr.goodbarber.com/creer-app/68-comment-creer-une-application-pour-smartphone) sans passer par le codage avec les langages natifs ? Il est possible aujourd’hui de créer des applications web puissantes (PWA). Elles sont créées à partir de langages web, et fonctionnent sur plusieurs systèmes d’exploitation.

Comment créer soi-même les applications natives ?

Connaître les langages natifs de programmation est indispensable pour créer soi-même une app. Ce sont Objective-C et Swift pour toute application publiée sur l’Apple Store. Les applications Google doivent être codées en Java ou Kotlin. C’est indispensable pour mobiliser les fonctionnalités natives. L’application dialogue avec le système d’exploitation dans le bon langage. Il faut toutefois se servir d’autres [outils pour créer une application Android](https://fr.goodbarber.com/creer-app/70-outils-pour-creer-application-android) .

Quels langages de programmation pour les applications Android ?

Le développement d’applications Android passe par la maîtrise du langage Java. Depuis juin 2019, Google a officialisé Kotlin comme langage à privilégier. Si vous souhaitez créer une application compatible avec plusieurs systèmes d’exploitation, pensez à l’application web, elle s’exécute sur les navigateurs. Elle fonctionnera avec Android et iOS indifféremment. [Développer une application mobile avec Python](https://fr.goodbarber.com/creer-app/69-developper-une-application-mobile-avec-python) est impossible, et déconseillé par les grandes acteurs (Apple et Google).

# Application Native, Hybride ou Web, comment faire son choix ?

Bien que très excitant et intéressant, le monde des applications mobiles n’est pas pour autant facile à cerner. Il existe différents types d’app, **ayant chacun leurs spécificités**. Après avoir [eu l’idée et avoir pensé à comment la transformer en application réussite](https://fr.yeeply.com/blog/convertir-idee-application-mobile/), il faut avoir en tête que chaque type d’application ne s’adresse pas forcément aux mêmes besoins, et c’est pourquoi il est important de bien cerner les différents types d’applications. Le but de cet article est donc de vous donner **une vision ainsi qu’une compréhension globale** de ce qu’est une application native, hybride ou web. La finalité est également de vous aider à faire votre choix. Il est primordial d’utiliser votre budget de façon optimale et en adéquation avec vos besoins. Après avoir eu l’idée et avoir pensé à comment la trasnformer en applciation réussite

Le vocabulaire des applications mobiles n’est pas familier pour tout le monde, et la barrière du langage technique peut poser de réels soucis. Une des questions que l’on reçoit souvent concerne la [différence entre iOS et Android](https://fr.yeeply.com/blog/developpement-dapplication-mobile-android-ios/). De nombreux termes peuvent limiter votre compréhension, et vous empêcher de faire un choix pertinent. Le plus important est de choisir ce qui est le plus approprié à votre projet.

## Les bases du monde des applications

Comme vous le savez sûrement, il existe deux systèmes d’exploitation qui dominent le marché : Android et iOS. Le premier est développé par Google et représente environ **80% du marché en France**. Tandis que son principal concurrent, iOS qui est développé par Apple, représente un peu moins de **20% du marché**. Vous l’aurez sûrement compris, ces deux systèmes d’exploitation s’adressent tous deux à des marchés distincts. Par exemple, les utilisateurs iOS sont en général plus aptes à dépenser de l’argent dans les apps et sont également plus exigeants. Pour cette raison, il est commun de trouver une application payante sur iOS qui sera gratuite sur Android. Les [business model](https://fr.yeeply.com/blog/12-business-model-du-developpement-d-applications/)  ne sont donc pas les mêmes pour ces deux plateformes.

On peut également comparer les deux systèmes d’exploitation mobiles avec des ordinateurs Windows et Mac. Une application Mac ne fonctionnera pas sur Windows et vice-versa. Les langages d’écritures sont différents, les règles sont différentes, les possibilités sont différentes et cela sans parler de la partie immergée de l’iceberg.

Aujourd’hui plus que jamais auparavant, les smartphones sont au cœur de nos vies et nous accompagnent au quotidien. Les applications que nous téléchargeons pour combler nos besoins confirment bien cette tendance. Concernant le marché des apps mobile, la domination de Google et Apple ne devrait pas s’arrêter de sitôt tant leur avance sur la concurrence est importante.

## L’application Native

Bien que le terme d’app native n’est pas très connu du grand public, cela **représente la grande majorité des applications que nous téléchargeons chaque jour**. Quand on pense aux applications, on pense inconsciemment aux applications natives. Mais alors, qu’est ce qu’une application native me direz-vous. C’est une application qui est développée spécifiquement pour un système d’exploitation. Je vous disais précédemment qu’une application Mac ne fonctionnerait pas sur Windows et vice-versa. L’explication est très simple : ce sont des équivalents d’applications natives.

Bien sûr, cela signifie que le langage de programmation est différent d’un système d’exploitation à un autre. Par exemple, iOS utilise majoritairement le langage Objective-C, tandis qu’Android utilise Java. Chacun a ses spécificités. C’est pourquoi vous verrez très souvent les développeurs préciser sur quelle plateforme ils développent. Un développeur Android ne sait pas  forcément développer sur iOS et vice-versa.

Passons maintenant aux choses sérieuses, les avantages de développer les applications en natif :

* **Meilleure rapidité**, fiabilité et dotée d’une meilleure réactivité ainsi qu’une résolution supérieure ce qui assure une meilleure [expérience utilisateur](https://fr.yeeply.com/blog/experience-utilisateur-ux-design/).
* **Elle permet un accès plus facile** à toutes les fonctionnalités du téléphone, de l’accéléromètre en passant par la caméra et même le micro.
* **Les notifications push**, uniquement disponibles sur les apps native. Ces notifications vous permettent d’alerter vos utilisateurs et d’attirer leur attention chaque fois que vous le souhaitez, que ce soit pour du nouveau contenu ou une offre promotionnelle.
* **Ne requiert pas forcément internet** pour fonctionner, ce qui est un réel avantage. Même en 2017, il existe encore des zones très peu couvertes par le réseau internet, et permettre à ses utilisateurs d’accéder à l’app sans connexion web est un très gros point fort à ne pas négliger.

Cependant, une app native ne permet pas seulement d’avoir des performances accrues et d’accéder à toutes les fonctionnalités du téléphone. Lorsqu’elles sont bien pensées et réalisées, elles respectent les codes design de chaque plateforme. Il existe des centaines de différences entre le système d’exploitation de Google et celui d’Apple. Les apps natives permettent de s’adapter à chaque plateforme afin de proposer à vos utilisateurs une expérience optimale.

## L’application Web

On a tous un jour fait face à un [application web](https://fr.yeeply.com/blog/6-types-developpement-d-applications-web/), sauf que l’on ne le savait peut-être pas. Par exemple, on a tous déjà fait face à une « version mobile » d’un site web. Vous l’aurez compris, la **version mobile d’un site web**, c’est une application web. À l’opposé de l’app native se trouve l’application web. Pourquoi à l’opposé ? Car c’est une application qui fonctionne comme un [template de site web](https://fr.yeeply.com/blog/web-design-vs-template-site-web/" \t "_blank), sans aucune différence de plateforme, de système d’exploitation, de code… Cela veut dire que **vos utilisateurs n’auront pas à installer l’application** sur leur smartphones, le stockage reste donc intact. Ce qui est un très gros point fort quand on voit à quel point la mémoire de nos téléphones se remplit vite. Entre les applications, les musiques et les photos, les gigas s’accumulent vite et on se retrouve très rapidement à gérer son stockage malgré nous.

En général, le but d’une application web est de **rendre du contenu disponible**, ou du moins fonctionnel sur mobile. Elle se comporte comme une application normale, ressemble à une application normale, mais son accessibilité et son prix n’ont rien à voir avec les applications mobiles. En effet, une application web coûte jusqu’à trois fois moins cher qu’une application native.

Image de Yeeply

Cependant, une application web a son lot d’inconvénients comme toute autre app. Elle ne permet pas d’utiliser les fonctionnalités au maximum de leur potentiel, et surtout, elle **ne fonctionne généralement pas sans internet**. Les applications web sont également plus lentes et moins adaptées à chaque plateforme. Elles ne sont pas développées spécifiquement pour une plateforme, ce qui limite leur adaptabilité. Le fait qu’une application web soit créée en une seule version pour toutes les plateformes et systèmes d’exploitation ne lui permet pas d’avoir des performances équivalentes à tout autre type d’application.

De plus, le fait que l’on ne puisse pas télécharger d’application web sur les stores rend le processus de fidélisation extrêmement compliqué. Sans compter le fait que cela **influe grandement sur le trafic de l’application** et donc son nombre de téléchargements. L’envoi de notifications push est également impossible avec une application web.

Bien sûr, tout n’est pas négatif avec les applications web, et les possibilités restent tout de même très larges. Il faut juste savoir que les cibles et objectifs de ce genre d’application sont différents des autres apps.

## L’application Hybride

Entre l’application native et l’application web se trouve l’application hybride. Comparées à une app native, elles sont plus rapides et moins chères à développer. Elles offrent également plus de possibilités qu’une app web. Au final cela donne un juste milieu, mais est-ce vraiment la meilleure solution ?

Un des principaux avantages de l’application hybride est qu’elle est **plus facile et plus rapide à développer qu’une app native**. La maintenance de l’application sera également plus facile puisqu’il n’y a qu’une seule version à revoir pour plusieurs plateformes. Cependant, toutes ces facilités ont un prix : les performances de l’application sont moins bonnes et moins stables puisque le système est moins adapté à chaque plateforme.

Bien sûr, il est possible de se rapprocher du fonctionnement d’une app native mais les coûts de développement s’en rapprocheront, sans avoir réellement le même niveau de qualité et de performance. Un des autres gros avantages du développement en hybride réside dans le fait que **vous gérez une seule version**, indépendamment du nombre de plateformes sur lesquelles vous voulez être présents. C’est-à-dire que si vous souhaitez ajouter de nouvelles fonctionnalités ou faire une mise à jour, vous n’aurez à **le faire qu’une seule fois**. Sur une application native, ces changements doivent se faire sur chaque plateforme. Cela prend plus de temps, et coûte également plus d’argent.

Prenons pour exemple la situation suivante : vous souhaitez développer une application à partir d’une site web ou d’une app web existante. Votre application web est simple et permet de faire tout ce dont vous avez besoin, mais elle n’a pas les spécificités d’une app native. C’est à dire qu’elle n’est pas présente sur les stores, vous ne pouvez pas envoyer de notifications push, ou encore le fait qu’elle ne soit pas présente sur l’écran d’accueil du smartphone de vos utilisateurs. Alors le développement en hybride sera sûrement **plus adapté, plus économique et plus simple** que le développement en natif. Cependant, il faudra garder en tête que vous serez **moins libre de faire ce que vous souhaitez**. C’est à dire que votre application devra rester simple. Pour toute application complexe, le développement en natif est de loin le meilleur.

## Au final, comment faire un choix ?

Vous allez être déçu mais il n’existe pas de réponse parfaite pour cette question. Il sera sûrement difficile pour vous de faire un choix. Mais cela s’explique par le nombre de facteurs à prendre en compte. Tout dépendra de vous, de vos besoins et objectifs, de vos moyens, des besoins de vos utilisateurs et surtout de la situation actuelle du projet. Vous souhaitez réaliser un jeu mobile ? Alors il faudra absolument le développer en natif. Les joueurs sont beaucoup plus exigeants au niveau des performances de l’application. Vous souhaitez créer une application simple et accessible par tous sans barrières de systèmes d’exploitation ? Alors l’application web sera plus adaptée à vos besoins. Enfin, vous souhaitez adapter votre app web au marché des apps mobiles ? Alors le développement en hybride sera sûrement plus adapté puisque peu coûteux et très rapide à mettre en place.

REACT NATIVE

Vous avez surement entendu parler de React Native, vous vous demandez si c’est un bon choix pour développer votre application mobile? Nous l’avons testé pour vous.

Dans cet article nous vous montrerons les Forces, Faiblesses, Opportunités et Menaces (SWOT) de chaque technologie. Mais avant tout quelques petites définitions:

## Qu’est-ce que ReactNative ?

React Native est un [framework d’applications mobiles open source](https://fr.wikipedia.org/wiki/React_Native" \t "_blank) créé par Facebook qui a vu le jour en 2015. Il permet aux développeurs de générer du code iOS et Android depuis un seul et même code maitre.

## Qu’est-ce que le développement natif ?

Un développement en natif implique deux codes indépendants: un pour iOS et un pour Android. Chacun étant écrit dans un langage dédié nativement pour l’OS (Système d’Exploitation) correspondant.

### **iOS**

#### **Objective-C**

C’est le langage historique pour développer sur iOS en natif. C’est un langage qui existe depuis bien avant la [sortie du premier iPhone en 2007](https://fr.wikipedia.org/wiki/IPhone) puisque la première version de ce langage date de 1983.

#### **Swift**

En [2014](https://fr.wikipedia.org/wiki/Swift_(langage_d%27Apple)), Apple lance la première version de son langage de programmation qui a connu une forte adoption du fait qu’il est plus simple qu’Objective-C. Aujourd’hui les développements natifs en iOS se font pour leur grande majorité en Swift.

### **Android**

#### **Java**

[Java est un langage de programmation](https://fr.wikipedia.org/wiki/Java_(langage)) qui existe depuis 1995. Il appartient à Oracle qui en 2012 annonce vouloir le mettre à disposition du développement des logiciels embarqués. C’est alors le principal langage de programmation utilisé pour développer des applications mobile Android.

#### **Kotlin**

Kotlin existe depuis 2011 mais c’est en 2017 que Google annonce lors de sa conférence Google I/O supporter ce langage pour les Applications Android. Ce langage reçoit alors une forte adoption. Découvrez [les 8 raisons pour lesquelles nous croyons en Kotlin pour développer les applications Android natives](https://www.kreactive.com/blog/tech/les-8-raisons-pour-lesquelles-nous-croyons-en-kotlin-pour-developper-les-applications-android-natives/).

-🡪 pas adapté, sert à faire du dev commun IOS / android, pas ce qu’on recherche

### **La différence entre hybride et cross-platform**

À des fins de simplification, j’utilise le terme “application hybride” pour parler à la fois des applications hybrides et des applications cross-platform mais il s’agit là d’un abus de langage. Les applications hybrides utilisent des éléments web pour créer l’application et son contenu puis un “wrapper” ou une enveloppe native pour que l’application web puisse être utilisée comme une application native sur le téléphone.

### **Applications web portées et PWA : Progressive WebApps**

Les applications “HTML5” qui ne mentionnent pas d’autres noms comme ceux cités dans cette section sont bien souvent des sites web portés sur mobile sont particulièrement limitées mais peuvent tout à fait convenir pour des applications simples comme une vitrine ou une publication périodique (magazine, blog, …).

### **Technologies hybrides : Ionic, Apache Cordova et PhoneGap**

Ces trois technologies sont toutes liées :

* [Apache Cordova](https://cordova.apache.org/) est un projet libre qui sert de base aux développements sur plusieurs plateformes en utilisant des technologies web (HTML5, Javascript et CSS) plutôt que du code natif
* [PhoneGap](https://phonegap.com/) est une version de Cordova produite et maintenue par Adobe
* [Ionic Framework](https://ionicframework.com/) est un ensemble de classes CSS et de bibliothèques et modules Javascript basé sur AngularJS, permettant d’augmenter Cordova

## La webapp

La webapp désigne la version mobile et responsive d’un site web. Par définition, elle ne nécessite donc pas un téléchargement préalable sur un appstore, elle est directement disponible via navigateur mobile en entrant l’adresse du site. Elle a pour avantage d’être moins coûteuse qu’une application native, mais aussi plus rapide à développer.

C’est ce que confirme Pierre, développeur chez Hellowork : « C’est effectivement plus rapide car il n’y a qu’un seul code pour les deux applications (Android & iOS). On se base sur des technos web qui sont maîtrisées par énormément de développeurs, il n’y a donc pas forcément besoin d’avoir un développeur purement mobile, un développeur front web peut s’y mettre. En contrepartie, elles offrent assez peu de contrôle sur l’aspect matériel, elles ne sont pas aussi fluides qu’une application native (même si les dernières technos hybrides comme ReactNative ou NativeScript offrent des performances très correctes). »

1 – Qu’appelle-t-on progressive web app (ou PWA) ?

Une progressive web app (ou PWA) est une application mobile développée avec des langages de développement web (HTML5, CSS3, Angular ou React) et exécutée depuis un navigateur web mobile. Son fonctionnement se veut très proche de ce que pourrait proposer une application classique (téléchargée depuis un store).

Il n’y a pas besoin d’installer une application et les mises à jour sont transparentes pour l’utilisateur. Cependant, elle ne peut fonctionner sans connexion Internet.

Les progressive web apps sont par exemple utilisées pour des versions de site web optimisées, ou des projets simples.

2 – Qu’est-ce que le développement natif ?

Une application native est développée dans le langage propre à la plateforme choisie et permet donc d’utiliser pleinement les capacités des appareils cibles.

Chaque grande plateforme a son ou ses propres langages :

* le Swift pour iOS (Objective C auparavant)
* le Java ou le Kotlin pour Android

L’utilisation de ces langages est différente et nécessite donc de faire appel à un développeur spécialisé.

Pour une application souhaitant être publiée à la fois sur l’Apple Store et le Play Store, il faudra donc multiplier le coût de développement, mais le résultat n’en sera que meilleur sur certains aspects :

* expérience utilisateur respectée pour chaque plateforme
* performance, puisqu’on pourra profiter de la pleine puissance de calcul du téléphone

3 – Et l’hybride dans tout ça ?

Une application hybride est à mi-chemin entre web app et application native. Elle se télécharge depuis un store, se lance comme une application classique mais s’exécute dans une web view (une page Internet chargée dans l’application).

Son développement s’effectue avec les langages du web et on peut choisir de porter cette application sur plusieurs plateformes à la fois (iOS et Android).

Elle permet un fonctionnement hors-ligne et offre la possibilité d’exploiter les capacités du système comme l’appareil photo, le GPS, les notifications, l’accéléromètre, le stockage de fichiers, etc.

4 – Comment faire son choix ?

Quelques éléments de comparaison dans le tableau ci-dessous :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Web app** | **Application native** | **Application hybride** |
| **Compatibilité sur les plateformes** | (+) Compatible sur tous les terminaux | (-) Compatible sur 1 seule plateforme | (+) Compatible sur tous les terminaux (nécessite une configuration particulière pour chacune) |
| **Expérience utilisateur** | (-) Limitée (même ergonomie et graphisme pour toutes les plateformes) | (+) Sur mesure | (-) Limitée (même ergonomie et graphisme pour toutes les plateformes) |
| **Performances** | (-) Réduites (dépend de la qualité du réseau et du navigateur utilisé) | (+) Optimales | (-) Réduites (dépend de la qualité du réseau) |
| **Hors connexion** | (-) Non disponible | (+) Disponible | (+) Disponible |
| **Distribution et mise à jour** | Via Internet (+) Mises à jour transparentes  (-) Fidélisation plus compliquée qui influe sur le trafic de l’application | Sur un store (-) Nécessite une re-publication et validation à chaque mise à jour (un peu long pour iOS) | Sur un store (+) Mises à jour transparentes pour la partie web |
| **Monétisation** | (-) Impossible de vendre le téléchargement de l’application  (+) Publicité | (+) L’application peut être payante sur le store (mais un pourcentage est prélevé)  (+) Achats in-app  (+) Publicité | (+) L’application peut être payante sur le store (mais un pourcentage est prélevé)  (+) Publicité |

Le coût de développement dépendra de la complexité du projet envisagé. Pour une interface mono-plateforme, le temps et le coût de développement seront assez proches, peu importe le type d’application.

A partir du moment où l’on voudra une application multi-plateforme, le coût du projet sera forcément moins élevé pour une web app ou une application hybride que pour des applications natives. Mais pour faire son choix, il faudra impérativement tenir compte des contraintes (fonctionnalités, ergonomie, fluidité, performance).

Progressive web app

Application

[C:\Users\J1016633\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\504C93A0.tmp](https://www.google.com/search?q=pwa&safe=active&tbm=isch&source=iu&ictx=1&fir=VzGQCncfAZawsM%252CoQNSAL8H34CShM%252C%252Fg%252F11bzxympx6&vet=1&usg=AI4_-kQRyHh2z5RQa9QDxvvgWVSgyFcVvg&sa=X&ved=2ahUKEwi4kKmqxLjqAhXrxoUKHcfqCrwQ_B16BAgUEA8#imgrc=VzGQCncfAZawsM)

## Description

## Description

Une progressive web app est une application web qui consiste en des pages ou des sites web, et qui peuvent apparaître à l'utilisateur de la même manière que les applications natives ou les applications mobiles.

**CONCLUSION :**

**Si WEB APP ou PWA : langages web + react (langage interprété)**

[**https://nimbleways.com/home/?p=7328**](https://nimbleways.com/home/?p=7328) **(pour le mode hors ligne)**

**Si Native : Kotlin / java (langage compilé)**