Choix technologiques

Développement d’applications natives :

Les langages pour créer des applications natives permettent d’avoir une application directement compilée sur le système cible. Les applications seront mieux optimisées et pourront accéder nativement aux composants physiques de l’appareil (caméra, micro…). Cependant, les langages ne permettent pas de créer une application cross-platform.

|  | Avantages | Inconvénients |
| --- | --- | --- |
| Java | * Langage que l’on connaît déjà tous * Facile à prendre en main * Grande variété d’outils déjà disponibles * Beaucoup de documentation en ligne | * Performances moins bonnes * Beaucoup de code pour peu de choses |
| Kotlin | * Se rapproche du Java * Compatible avec java * Recommandé par Google pour ses fonctionnalités modernes * Fonctionnalités supplémentaires par rapport à Java (meilleure gestion des threads… * IntelliJ prend en charge Kotlin | * Moins de documentation * Peu de code pour beaucoup de choses * Compilation plutôt lente * Adaptation avec le langage à prévoir |
|  |  |  |
|  |  |  |

Développement d’applications cross-platform (et hybride pour Cordova) :

Les langages de développement cross-platform permettent de créer des

applications compatibles avec des plateformes différentes (iOS ou Android).

Pour le développement hybride, c’est utile pour des applications simples mais au détriment de l’expérience utilisateur et de la performance.

|  | Avantages | Inconvénients |
| --- | --- | --- |
| Dart (+flutter) | * Haute performance (exécutable sous formes différentes, en javaScript, en code natif, anticipée…) * Adaptable en application web * Syntaxe simple (orienté objet, typage de variables) * Hot reload (Vue en temps réel sur un émulateur des changements dans le code) * Nouveau langage à découvrir et rajouter sur le CV * Le framework Flutter permet de créer des applications adaptées à l’environnement mobile et personnalisées * Performances proches des langages à applications natives | * Langage jeune donc moins de développeurs dans ce domaine donc peut-être moins d’aide * Nouveau langage donc nécessité d’apprendre tout comme Flutter * Applications plus lourdes * Accès limité à certaines API |
| JavaScript/TypeScript (+ React Native) | * Langage très populaire * Beaucoup de documentation * Hot Reload * Bonne expérience utilisateur * Langage que l’on connaît déjà et/ou facile à apprendre * Outils de développement nombreux | * Haute complexité lorsque l’application devient plus complexe * Syntaxe bordélique * Fonctionnement aléatoire * Peut nécessiter des extensions supplémentaires * Moins performant dans certains cas |
| Cordova | * Utilisation de langages web pour créer l’application (HTML/CSS/JavaScript) * Langages communs * Beaucoup de documentation dans ces langages * Beaucoup de plugins pour accéder à des composants du mobile | * Outils lent et buggés (par mon expérience) * Performance limitée * Expérience utilisateur moins bonne (étant donné que c’est du web à la base) * Accès limité à certaines fonctionnalités |