

# Étude comparative des différentes technologies côté client

---

CODEV

BENALI Myriam – NAAJI Dorian  
POLYTECH LYON  
5A INFO GROUPE 2, ÉQUIPAGE 2

# 1.INTRODUCTION

Ce document a pour but de préciser ce qui a motivé nos choix technologiques pour la partie cliente de l'application M' Ton Air, sachant que nous souhaitons **de préférence** réaliser une application hybride pouvant être déployée à la fois sur mobile (web et application) et ordinateur (web).

## 2.LES TECHNOLOGIES

### 2.1. React Native

React Native est un framework javascript qui permet de construire des applications natives pour mobiles Android/iOs/Windows Phone en écrivant du code une seule fois pour les deux. L'accès aux fonctionnalités embarquées du mobile (localisation, appareil photo) est simplifié.

Le framework permet de manipuler des composants et éléments d'interface.

### 2.2. Ionic

Ionic contrairement à React Native permet de développer une application web qui est ensuite encapsulée dans un « wrapper », pour déployer l'application sur mobiles Android/iOs. Cela impacte entre autres l'expérience utilisateur et les performances de l'applications, qui seront toutes deux en deçà de ce que pourrait proposer un framework « natif ». Mais cela permet de coder une seule fois et d'avoir de la portabilité sur web, mobiles, etc.

Néanmoins, Ionic permet de développer en HTML/CSS/JavaScript et donc il y a « moins de choses à apprendre ».

### 2.3. Flutter

Flutter permet de développer des applications natives tout comme React Native. Le framework, développé par Google, est encore très peu utilisé dans le monde de l'entreprise.

Il est nécessaire également d'apprendre un autre langage, Dart, afin de pouvoir s'en servir. Dans l'ensemble, nous avons décidé de mettre cette technologie de côté pour ces raisons. Néanmoins, l'application propose des composants d'interface très robustes et esthétiques. Le langage Dart corrige de plus certaines imperfections de JavaScript.

## 2.4. Vue Native

Vue Native est un framework cross-platform pour construire des applications natives en utilisant JavaScript. Le framework permet de connecter React Native et Vue.js, une librairie JavaScript permettant de manipuler des interfaces mobile.

Étant donné que Vue est ensuite utilisé par React, nous mettrons ce framework de côté.

## 3. CONCLUSION

Dans une application mobile, tout se joue sur l'expérience utilisateur. Si l'application ne plaît pas, elle sera aussitôt désinstallée. Ce point de vue doit être fortement pris en compte dans le développement de l'application.

D'après plusieurs sources, un framework pour construire des applications natives rendrait l'expérience utilisateur plus agréable et augmenterait dans l'ensemble les performances. Facebook par le passé a utilisé du HTML/CSS/JavaScript grâce à une application hybride, et lors du passage au natif, l'application a vu sa popularité croître.

D'après ces quelques éléments, notre choix se porte donc vers le natif. Après comparaisons entre les frameworks proposant du natif, nous allons nous pencher vers l'utilisation de React Native. Ce framework est très utilisé dans le monde de l'entreprise, très « à la mode » et peut être une réelle valeur ajoutée à un CV. Outre cela, nous avons l'impression que beaucoup d'autres frameworks comme Vue Native reposent ensuite sur React Native.

En ayant comparé les 4 précédentes solutions, notre choix sera donc d'utiliser React Native.

## 4. SOURCES

[\[1\] : Ymedialabs : Hybrid vs Native](#)

[\[2\] : Ionic Framework](#)

[\[3\] : React Native](#)