[**https://alkalmazasfejlesztes.blog.hu/2019/05/22/mobil\_alkalmazas\_fejlesztes\_nativ\_cagy\_cross\_platform**](https://alkalmazasfejlesztes.blog.hu/2019/05/22/mobil_alkalmazas_fejlesztes_nativ_cagy_cross_platform)

[**Natív vagy cross-platform?**](https://alkalmazasfejlesztes.blog.hu/2019/05/22/mobil_alkalmazas_fejlesztes_nativ_cagy_cross_platform)

## Mikor beszélhetünk natív fejlesztésről?

Mobil applikáció készítése esetén natív applikáció fejlesztésnek nevezzük, amikor az adott mobil platformhoz **hivatalosan kiadott fejlesztői környezetet és programozási nyelvet** használjuk.

Amennyiben Andriod eszközökről van szó, a következők tekinthetők natív fejlesztésről:

* Google által kiadott [Android Studio fejlesztői környezet](https://developer.android.com/studio) mellett
* [Java](https://codelabs.developers.google.com/codelabs/build-your-first-android-app/#0), vagy
* [Kotlin](https://kotlinlang.org/docs/tutorials/kotlin-android.html) programnyelv alklamazása.

Hasonlóan, iOS esetén akkor beszélhetünk natív applikáció fejlesztésről, ha:

* az Apple által kiadott[Xcode fejlesztői környezet](https://developer.apple.com/xcode/)et és
* [Objective-C](https://developer.apple.com/library/archive/documentation/General/Conceptual/DevPedia-CocoaCore/ObjectiveC.html), vagy
* [Swift](https://developer.apple.com/swift/) programozási nyelvet használunk.

## Mit nevezünk cross-platform technológiának?

[Cross-platform technológiának](https://en.wikipedia.org/wiki/Cross-platform_software) azt nevezzük amikor egyetlen kódbázisból "egy fejlesztéssel" több platformra is elkészíthetjük alkalmazásunkat. A cross-platform technológiákból léteznek kiemelkedő teljesítményű rendszerek, melyekkel a natív appokkal megegyező felhasználói élményt nyújthatunk, de vannak kevésbé jól felhasználható keretrendszerek is. A cross-platform applikációknak általánosságban azonban van létjogosultságuk, mivel nagyban csökkentik a fejlesztés költség- és időigényét. Az egyes cross-platform megoldások közt azonban jelentős eltérések lehetnek a megvalósítható funkciókat illetően. Mivel a régebbiek főként webes technológián alapulnak, így nem minden funkció fejleszthető le az alkalmazásban. A modernebb cross-platform technológiák esetén a cross-platform kód és a natív kód között már van átjárhatóság. Így ha egy részletet nem lehet megoldani közös kódbázisból, lehetőség van  annak megírására a platform saját nyelvén és környezetében.

## Mik a legelterjedtebb cross-platform keretrendszerek?

### React Native

A[React Native](https://facebook.github.io/react-native/) az egyik legelterjedtebb cross-platform keretrendszer, segítségével React és JavaScript technológiákkal készíthetünk mobil alkalmazásokat. A React Native komponensek natív ekvivalensükké fordulnak, ami a teljesítmény szempontjából igen nagy előnyt jelent a többi cross-platform keretrendszerrel szemben. Emellett a React Native lehetőséget biztosít a natív kódrészletek felhasználására is, így amennyiben egy probléma nem megoldható tisztán a keretrendszer segítségével, úgy a fejlesztők natív kóddal (Swift, vagy Objective-C iOS esetén és Java, vagy Kotlin Android esetén) egészíthetik ki a cross-platform kódot. A React Native egy, a Facebook által támogatott, nyílt forráskódú és felhasználású projekt. A Facebook mellett azonban már számos más területen működő vállalat és startup is használja a mobil applikációjához a React Native-et. A teljesség igénye nélkül néhány alkalmazás, amelyet React Native-ben fejlesztettek.

* Facebook
* Instagram
* Skype
* Tesla
* Pinterest
* Uber

### Ionic

Az[Ionic](https://ionicframework.com/) hibrid keretrendszer szintén kedvelt választás a cross-platform eljárást preferálók körében. Mivel az Ionic javarészt webtechnológián alapul, így egyetlen kódbázisból fejleszthetünk Androidra, iOS-re és webre is. Ez a gyakorlatban sajnos számos hátrányt is von magával, a webtechnológiák használata valamivel gyengébb teljesítményt nyújt a natív, vagy a natívra forduló (pl. React Native) cross-platform keretrendszerekkel szemben. Emellett sokáig az Ionic ellen szólt, hogy nem tette lehetővé natív kód felhasználását. A közelmúltban az Ionic Native könyvtár bevezetésével válaszolt az Ionic közössége az említett igényre. Néhány applikáció, amit Ionic segítségével készítettek:

* Pacifica
* MarketWatch
* Sworkit
* Nationwide

### Flutter

A [Flutte](https://flutter.dev/)r keretrendszer a Google cross-platform megoldása, a React Native-hez hasonlóan natív komponensekre fordul, a natív appokkal megegyező teljesítményt biztosít. A [Flutter](https://en.wikipedia.org/wiki/Flutter_(software)) egy viszonylag fiatal technológia, az alfa verzió 2017 májusában, a stabil kiadás pedig 2018 decemberében jelent meg. Ennek megfelelően kevés helyen használják éles alkalmazásonál azonban már így is olyan appokat készítenek a technológiával mint a Google Ads vagy az Alibaba. A Flutter technológiára érdemes lehet figyelni, a közeljövőben felveheti a versenyt a React Native alapú technológiákkal.