**Ionic или React Native: что лучше для стартапа**

С какой дилеммой сталкиваются начинающие разработчики мобильных приложений? Самый наболевший вопрос для них – выбор фреймворка для реализации своей идеи. Одни склоняются к Ionic, другие готовы с мышкой в руках отстаивать честь React Native. Но мы постараемся быть объективными и рассмотрим все за и против для каждого варианта!

**Что такое Ionic и какие у него особенности**

Ionic представляет собой SDK с открытым исходным кодом. С его помощью можно создавать гибридные приложения для Android, iOS и даже Windows. Сам фреймворк интегрируется в CSS, HTML и может взаимодействовать с React, Angular или Vue.

Преимущества Ionic:

* Возможность экспресс-разработки и тестирования.
* Один и тот же код можно задействовать для разных платформ.
* Проведение тестов в любых браузерах, без эмулятора.
* Встроенные стили UI-компонентов с доступом из фреймворка.
* Обширный список сторонних библиотек на Angular и JavaScript.
* Наличие Pro Development Environment, ускоряющего разработку.

Недостатки Ionic:

* Созданные в нем приложения работают медленнее нативных.
* Каждая из версий фреймворка содержит те или иные баги.

С помощью Ionic были разработаны приложения Marketwatch, Nationwide, Pacifica, Untappd, JustWatch и другие.

**Что такое React Native и с чем его «едят»**

React Native является JavaScript-фреймворком, посредством которого можно создавать похожие на нативные приложения для Android и iOS. Они разрабатываются на смеси, состоящей из знакомого многим языка JavaScript и разметки XML. А еще фреймворк может похвастать наличием моста, который позволяет заниматься нативным рендерингом API в Java для решений на Андроид и в Objective-C для приложений на iOS.

Преимущества React Native:

* Отличная производительность благодаря работе на собственном API.
* Возможность обновления приложения без закрытия текущей версии.
* Можно использовать свои модули в Java, Objective-C, Kotlin и Swift.
* Взаимосвязь с «родными» контроллерами UI и поддержка ReactJS.
* Огромное сообщество разработчиков и созданных ими библиотек.

Недостатки React Native:

* Необходимость ручной адаптации приложения для другой платформы.
* Длительное время разработки, если сравнивать с тем же Ionic.
* Тяжеловесные приложения, чей размер больше, чем на Kotlin или Java.
* Подгонка встроенных компонентов с выбором виджетов и стилей.
* Возможно, придется привлекать сторонних программистов и инвестиции.

С использованием React Native были разработаны приложения Instagram, Airbnb, Facebook, Bloomberg: Finance Market News, UberEats и другие. Правда, часть из обозначенных решений позднее перестали использовать данный фреймворк.

**Какой из фреймворков будет лучше для стартапа**

Предлагаем провести «очную ставку», то есть небольшое сравнение Ionic и React Native.

Чтобы создать свое приложение в Ionic, достаточно владеть азами JavaScript и HTML со стилями CSS. Также понадобится небольшой опыт в одном из фреймворков, например, в React, Angular или Vue. Углубленные познания в данных технологических стеках не обязательны, будет достаточно базового уровня. А для разработки в React Native без совершенного владения React.js и JavaScript не обойтись. То же самое можно сказать и о Swift, Kotlin, Objective-C и Java – все эти языки понадобятся любителям «натива».

В Ionic один и тот же код можно задействовать для любых гаджетов и операционных систем. Что касается React Native, тут ситуация похожая, но менее универсальная. Чтобы приложение работало как надо, может потребоваться дополнительный фрагмент кода, который обеспечит необходимый уровень функциональности для конкретной платформы.

После создания приложения его следует проверить. Ionic позволяет выполнить тестирование ПО прямо в браузере. А React Native обязывает применять эмулятор.

Обновление приложения, созданного на Ionic, осуществляется лишь после внесения изменений. В React Native это происходит в режиме реального времени.

Производительность решений на Ionic уступает скорости работы приложений на React Native. Разница не сильно заметна, но в некоторых сценариях может быть ощутимой.

Вы можете выбрать React Native, если вы или ваши коллеги в нем хорошо разбираются. Также такой выбор будет оправдан в случае, если вы готовы уделить разработке приложения много времени, вложить денежные средства на оплату труда специалистов со знанием других языков программирования и у вас в приоритете – производительность.

А если ваш бюджет ограничен, вы хотите обойтись своими силами, но не готовы при этом тратить много времени, желаете сосредоточиться на создании MVP, выбирайте Ionic!