

Flutter peut-il être utilisé dans un cadre professionnel?

Pour créer des applications, il existe plusieurs solutions : il y a la méthode native et les frameworks multiplateforme. Pour cette deuxième solution, qui réduit considérablement les coûts de développement de l'entreprise, il y a la possibilité de partir sur du web, ou bien d'utiliser un "vrai" framework multiplateforme. Parmi tous ces frameworks, il en existe un qui s'appelle Flutter, qui a été développé par Google et est sorti en 2017.

De base, ce framework est utilisé pour tout ce qui est interface utilisateur. Cependant, depuis quelques temps, il est surtout utilisé pour sa capacité à concevoir des applications natives multiplateforme, notamment pour Android et iOS. Le langage utilisé par ce framework est Dart, qui offre des grands avantages pour les utilisateurs comme pour les programmeurs.

En effet, un code créé avec du Dart sera optimisé directement par rapport au support sur lequel il fonctionnera grâce au mode de fonctionnement "AOT" (Ahead Of Time), ce qui est un grand avantage pour l'utilisateur, car si une application est optimisée, cela veut dire qu'elle sera plus rapide. Et de plus, Dart est un langage assez performant par lui-même.

Il offre aussi un avantage pour les développeurs, qui est celui de diminuer le temps entre chaque build. En effet, ce temps (qui peut être horriblement long dans certains cas) est diminué grâce à la fonctionnalité "Hot Reload". Or, si on réduit ces temps entre les build, on diminue aussi le temps passé à développer une application et on optimise le temps de développement.

Il y a un autre avantage non négligeable à Flutter : Google ayant recréer l'ensemble des composants graphiques en Flutter, les développeurs sont certains d'avoir toujours le même rendu quelle que soit la plateforme sur laquelle sera déployée l'application.

Enfin, il y a aussi le nombre incroyablement élevé de Widgets que propose Flutter. L'idée du Widget est qu'un Widget spécifique ne doit faire qu'une seule et unique tâche. Par exemple le Widget ListView ne saura qu'afficher une liste à l'écran, son positionnement ou sa couleur sera géré par d'autres Widgets. Il existe d'ailleurs deux types de Widgets : les Widgets sans état et les Widgets avec état, ce qui ressemble aux différents types de composants présents dans React.

Pour l'instant on pourrait croire que Flutter est l'application miracle. Mais il a aussi ses défauts.

Tout d'abord on ne peut pas télécharger les applications développées avec Flutter sous iOS, il faut donc passer par un autre moyen, comme par exemple Flutter Gallery, afin que l'utilisateur puisse compiler par lui-même ce genre d'application.

De plus, Flutter est plus récent que son plus sérieux concurrent, React-Natif, ce qui fait que Flutter a moins de paquets disponibles comparé à ce dernier. Mais il faut noter que Flutter rattrape petit à petit son retard dans ce domaine, et qu'il n'est pas impossible que Flutter atteigne voire rattrape son concurrent. Il faut aussi noter que

En outre, sa prise en main est assez compliquée de par le fait qu'il faille apprendre le langage Dart, qui n'est généralement pas déjà connu par le développeur. En outre, la programmation réactive est aussi assez compliquée pour des personnes non averties. Ces deux aspects-là peuvent rebuter les développeurs aux premiers abords, surtout lorsque son concurrent, React-Natif, a une prise en main beaucoup plus facile puisqu'il utilise du JavaScript, qui est un langage beaucoup plus commun.

Conclusion

Flutter est un framework qui possède beaucoup de points forts, même s'il possède aussi son lot de faiblesses. Mais ses points faibles sont surtout des problèmes de prise en main, ce qui fait que si on prend le temps, Flutter est vraiment un excellent framework pour développer des applications multiplateformes. Pour terminer, on peut relever que Flutter est utilisé par des entreprises reconnues, comme Alibaba, le géant chinois, ce qui nous montre sa fiabilité et solidité. On peut donc lui prévoir une utilisation plus importante pour les futures années.