

Environnement .Net et C# IOS et Android

Faneva Ramiandrisoa

EFREI - Caplogy

12 Septembre 2022



- 30 Heures
- Mobile : Android
- C# et .Net
- Modalité d'examen :
 - Petit Projet Android ou C# .Net

Développement Mobile: Android

- Historique
- Vue d'ensemble
- Acteurs économiques
- Systèmes d'Exploitation Mobiles
- Développement Mobile
- Développement Android

Historique

- 1908, un brevet déposé à Kentucky pour un téléphone sans fil
- 1946, premier appel sans fil à partir d'un camion
- 1947, premier centre d'appels mobiles (5000 clients, 30000 appels par semaine)
- 1965, premier téléphone sans fil inventé
- 1983, premier appel mobile commercial
- 1992, sortie du 1er PDA par Apple
- 1994, invention du bluetooth
- 1997, première photo prise par un mobile
- 2003, 1er smartphone blackberry
- 2007, Création de l'iPhone
- 2008, première plateforme de paiement mobile
- 2009, Lancement de l'android market
- 2010, Lancement de Windows Phone store

Vue d'ensemble

- Milliards d'utilisation
- 1 smartphone pour chaque 5 personnes (en 2017)
- 89% des utilisateurs de smartphones l'utilisent chaque jour

Acteurs économiques

- Créée en 1976
- Entrée dans le marché en 2007
- Hardware : iPhone — iPad
- Software : iOS — macOS
- Cloud : iCloud



Acteurs économiques

- Créée en 1998
- Entrée dans le marché en 2008
- Hardware : Nexus
- Software : Android
- Cloud : Google Drive — Gmail



Acteurs économiques

- Créée en 1975
- Entrée dans le marché en 2001
- Hardware : Surface — Nokia
- Software : Windows 8 — Windows 8 RT — Windows Phone
- Cloud : SkyDrive — Office 365 — Outlook



Microsoft

Acteurs économiques

- Créée en 1969
- Entrée dans le marché en 2001
- Hardware : Galaxy SX — Galaxy Tab...
- Software : Tizen — Bada

The Samsung logo is displayed in a large, bold, blue sans-serif font, centered within a light gray rectangular background.

Acteurs économiques

- Black Berry
- HTC
- Huawei
- Asus
- Nokia
- Amazon
- LG
- ...

Systèmes d'Exploitation Mobiles

- Equipe un smartphone ou une tablette
- Plusieurs similarités avec les OS classiques
- Fonctions de réseaux et de téléphonie
- Tactile
- Photo et Vidéo
- GPS
- ...

- Equipe iPhone, iPad, iPod Touch et Apple TV
- Sorti en 2007 sous le nom de iPhone OS
- Exclusivité pour les produits Apple
- Tactile et multitouche
- App store ouvert en 2008

Android

- Open Source avec quelques composants propriétaires
- Écrit en C / C++ / Java
- Développé par Google et OHA
- Annoncé en 2007
- Premier dispositif en 2008
- Cible les smartphones, tablettes, TV connectées et voitures
- Plusieurs constructeurs dont Samsung et Motorola
- Google Play

- Windows Phone
- Bada
- Tizen
- Symbian
- BlackBerry 10
- WebOS
- Firefox OS
- Ubuntu Touch
- ...

Développement Mobile

Développement Mobile

- Développement webapp (Site mobile)
- Développement hybride
- Développement multiplateformes (crossplatform)
- Développement native

Développement webapp (Site mobile)

- Avantages

- Budget réduit et mutualisation des développements web
- Temps de mise en place relativement rapide
- Maintenance et mise à jour plus souple
- Parc cible important
- Montée en compétence rapide des équipes projet
- Pas besoin des stores

- Inconvénients

- Connexion réseau indispensable
- Accès limité aux fonctionnalités " natives" des systèmes
- Temps de chargement à l'utilisation
- Performances limitées
- Accès moins intuitif qu'une application

- Avantages

- Simplicité des développements
- Un seul “code” déployé sur plusieurs plateformes
- Temps de mise en place rapide
- Maintenance plus souple
- Coûts de développement réduits
- Environnement de développement simplifié (technologies web)

- Inconvénients

- Performances très limitées
- Différences d'affichage et de comportement potentiel selon la plateforme d'exécution difficiles à anticiper
- Pas d'accès à l'ensemble des fonctionnalités natives
- Outils de développement peu adaptés à des projets complexes
- Évolution limitée

Développement multiplateformes (crossplatform)

● Avantages

- Partage possible des développements indépendamment des environnements cibles
- Un seul code à maintenir, donc plus souple à maintenir et à faire évoluer
- Équipe de développement réduite
- Meilleur rapport Performances/Rapidité de développement
- Possibilité d'accès à quasiment toutes les fonctionnalités natives

● Inconvénients

- Obligation de prendre en compte les spécificité du matériel dans une certaine limite
- Outils de développements généralement propriétaires
- Outils de développements contraignants
- Lenteur relative des mises à jour des solutions de développement par rapport à l'évolution des matériels et des systèmes
- Communauté de développeurs plus réduite



- Avantages


- Performances des applications
- Accès à toutes les fonctionnalités natives et spécifiques du matériel
- Au plus près de l'ergonomie et des conventions habituellement utilisée par les utilisateurs sur leur smartphone
- Outils de développement adaptés au mieux aux plateformes et évoluant avec le matériel
- Adapté à des projets complexes et/ou de taille importante et disposant d'un long cycle de vie


- Inconvénients

- Capitalisation complexe des développements entre plateformes
- Processus de développement lourds et peu flexibles
- Coûts incompressibles importants surtout pour des petits projets
- Équipes de développement potentiellement plus nombreuses (une équipe par plateforme)


Comparaison

Une application
NATIVE, CROSS-PLATFORM, HYBRIDE :
COMMENT CHOISIR ?






VS



VS




Critères	NATIVE	HYBRIDE	CROSS PLATFORM
Performance élevée	✓	✗	✓
Plein usage des fonctionnalités du terminal mobile	✓	✗	✗
Budget moindre à l'achat	✗	✓	✓
Expérience utilisateur optimisée pour chaque OS	✓	✓*	✓*
Maintenabilité, évolutivité	✓	✓**	✓**
Planning serré de mise à disposition	✗	✓	✓

* Meilleure expérience utilisateur pour adopter l'application standard de navigation spécifique à chaque OS
** Meilleure maintenance de versions plus fréquente


COÛT GLOBAL DE L'APPLICATION (DÉVELOPPEMENT + MAINTENANCE)

NATIF




Développement de l'application

HYBRIDE



Maintenance de l'application

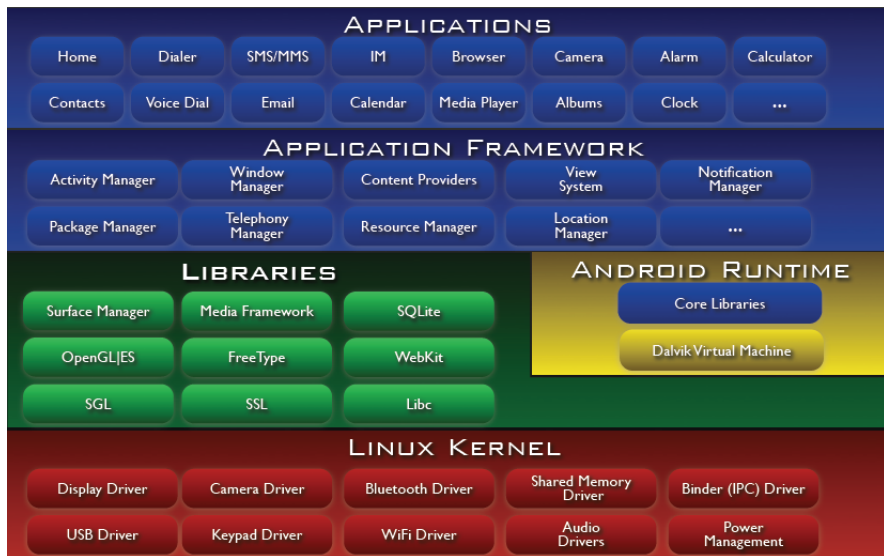
HYBRIDE NATIF



Coût total sur le long terme

Développement Android

Architecture



Composant d'application

- Vues
- Activités
- Fragments
- Intentions
- Fournisseurs de Contenu
- Services
- Notifications
- Ressources