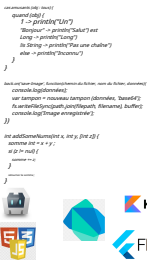


INF4188 : Développement du Mobile Applications

Environnements de développement de téléphones mobiles



Azanzi Jiomekong

Université de Yaoundé I, Département d'informatique

28 octobre 2022



Copyright (c) 2013 Jiomekong Azanzi Fidèl.

L'autorisation est accordée de copier, distribuer et/ou modifier ce document selon les termes de la licence de documentation libre GNU, version 1.2 ou toute version ultérieure publiée par la Free Software Foundation ; sans sections invariantes, sans textes de couverture et sans textes de couverture arrière. Plus d'informations sur la licence sont incluses dans la section intitulée « La licence de documentation gratuite GNU ».



Ce support est en construction. Pour m'aider à l'améliorer, merci d'envoyer vos commentaires à jiofidelus@gmail.com .

Objectifs (1/2)



A l'issue de ce cours, l'étudiant devra connaître :

- Technologies et outils pour le développement d'applications pour téléphones mobiles
- Cadres pour amorcer le développement d'applications mobiles
- Conseils sur les technologies à adopter lors du développement d'applications pour téléphones mobiles

Objectifs (2/2)



A l'issue de ce cours, l'étudiant devra connaître :

- Mettre en place un environnement de développement Android
- Configurer un environnement de développement iOS
- Mettre en place un environnement de développement mobile basé sur le Web
- Mettre en place un environnement de développement mobile multiplateforme
- Mettre en place un environnement de développement mobile basé sur une compilation



- Types d'applications mobiles
- Applications mobiles natives
- Application Web mobile
- Application mobile multiplateforme
- Application mobile compilée

Tout d'application mobile



- Natif : les applications sont écrites dans une langue native
- Web Mobile : sites Web pour appareils mobiles
- Hybride : aussi appelée application mobile Crossplatform, c'est la combinaison de Native et Web app
- Compilé : applications créées et compilées dans un code natif

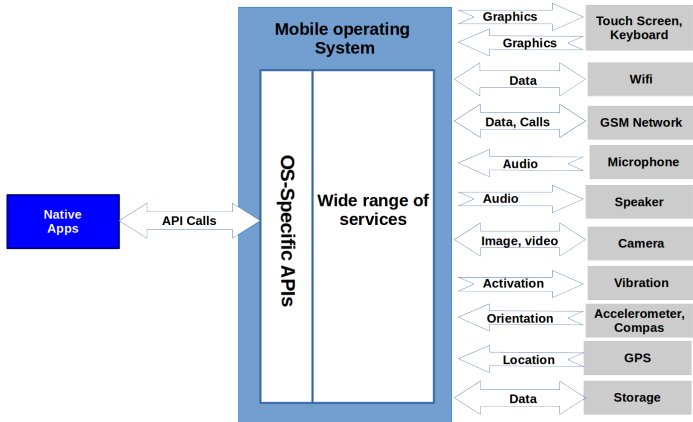


Native applications mobiles

- Type d'application mobile le plus courant
- Expériences d'interface utilisateur riches
- Construire pour une plate-forme spécifique
- Écrit l'application dans la langue native de la plate-forme donnée
- Android fournit un framework basé sur le langage Java
- iOS fournit un framework natif basé sur le langage Objective-C/
Swift

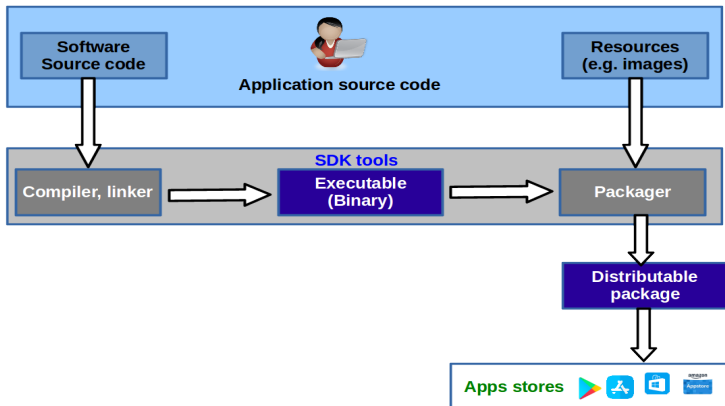
Native applications mobiles

Interaction avec le téléphone



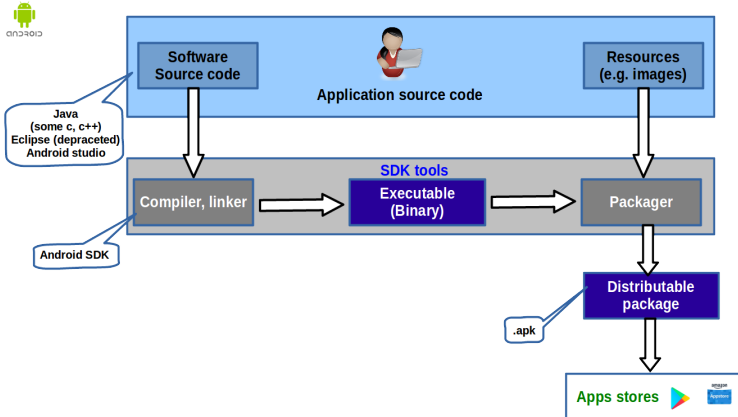
Nactife applications mobiles

Processus de développement



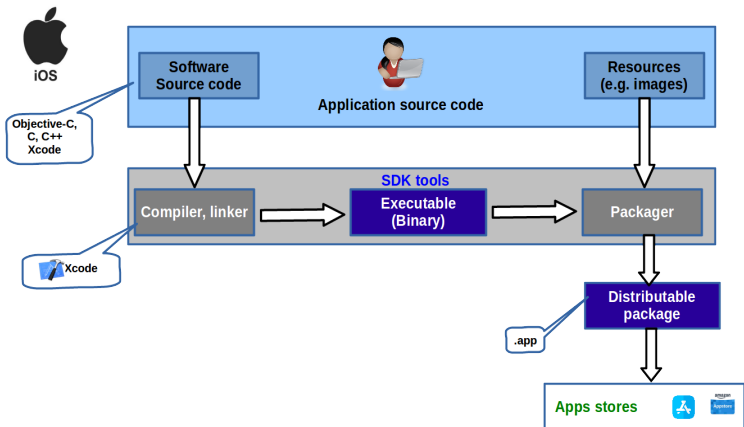
Nactife applications mobiles

Android



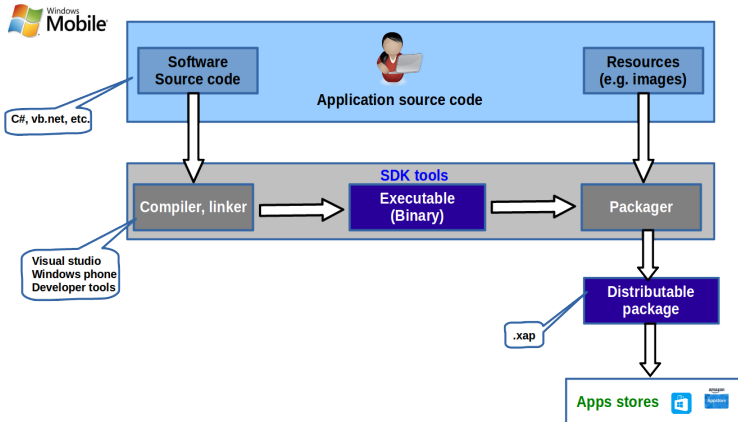
Nactife applications mobiles

IOS



Native applications mobiles

Windows Mobile





Native applications mobiles

Technologies clés

Android:

- Langage de programmation : Java, Android
- Kit de développement logiciel (SDK) : JDK, SDK Android
- Environnement de développement intégré : Eclipse (obsolète), Android Studio

IOS :

- Langage de programmation : Objective C, Swift
- Kit de développement logiciel (SDK) :
- Environnement de développement intégré : XCode



- Système d'exploitation : Microsoft Windows, Linux, Mac OS
- SDK : JDK 5, SDK Android
- IDE : Android Studio, IntelliJ IDEA, etc.
- Simulateurs Android : certains sont installés sur Android SDK

Unroïde

Outils de développement



titanium



Android NDK

- Téléchargez et installez le JDK
- Téléchargez et installez le SDK Android
- Définissez les variables d'environnement PATH et JAVA HOME pour faire référence à la bibliothèque
- Téléchargez et installez Android Studio



- Appareil : tout appareil Apple tel que MacBook Pro, Mac Mini
- Système d'exploitation : système d'exploitation OS X
- IDE : Xcode
- Compte développeur Apple
- Simulateur : iPhone/iPad virtuel, installé
- Un véritable appareil pour utiliser certaines fonctionnalités de l'appareil



Communauté de développeurs iOS

- Vidéos de la Conférence mondiale des développeurs (WWDC)
- Rejoignez les équipes du programme de développement iOS lorsque vous y êtes invité.
- Inscrivez-vous au programme pour développeurs iOS : aide les développeurs à fournir des applications de qualité aux utilisateurs

Avantages de rejoindre le programme développeur :

- Exécutez les applications que vous développez sur le véritable appareil iOS.
- Distribuez les applications sur l'App Store.
- Accédez aux aperçus des développeurs.



- XCode comprend tout pour développer des applications OS X, Watch OS, TV OS
- Pour l'utiliser, develop doit avoir :
 - Un appareil exécutant OS X
 - **SDK**
 - Un identifiant Apple
- Le simulateur iOS se compose de deux types d'appareils :
 - Conception iPhone avec différentes versions
 - Appareil iPad avec différentes versions



Native applications mobiles

Avantages désavantages

Avantages :

- Une voie éprouvée pour le développement mobile
- Très vite
- Boutiques d'applications distribuées : Google Play, Windows Store, ...
- Peut être approuvé très facilement
- Interactif et intuitif
- Interagir avec les utilitaires des appareils

Désavantages:

- Plateforme unique
- Un langage plus dur
- Très cher
- Difficile à entretenir

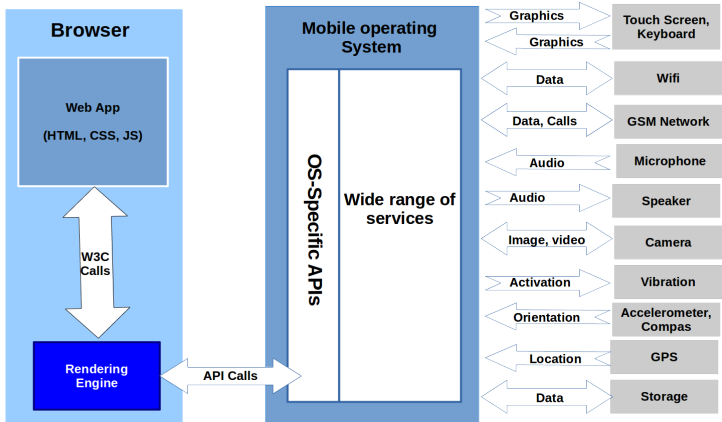
Web Mo développement d'applications biliaries



- Construisez comme un site Web ou une application Web classique
- Fonctionne dans un navigateur

Web Mo développement d'applications biliaries

Interaction avec le téléphone





Web Mo développement d'applications biliaries

Technologies clés

- Langage de programmation : HTML, CSS, JS
- Frameworks : NodeJS, Sencha Touch, Ratchet, JQuery mobile, Angular, React, VueJS
- IDE : Eclipse, Netbeans, Visual Studio Code, Sublime Text, etc.
- Tests : téléphone mobile lui-même, outil de développement Chrome, Riple
- Bibliothèques et frameworks CSS utiles :
 - Bootstrap, Materialise, Bulma, UIKit, Meteor, Foundation, interface utilisateur sémantique
 - animate.css, imagehover.css, chargement.io, soupçon.css, squelette.css, mini.css, police géniale

J.Query Mobile



- Cadre d'interface utilisateur open source
- Construit sur le noyau jQuery et développé par l'équipe du projet jQuery en 2010
- Utilisé pour développer des sites Web/applications réactifs
- Les applications développées sont accessibles sur les appareils mobiles, tablettes et ordinateurs de bureau
- Utilise les fonctionnalités de jQuery et jQueryUI pour fournir des fonctionnalités API pour les applications Web mobiles
- Fournit un ensemble d'entrées de formulaire tactiles et de widgets d'interface utilisateur



J.Query Mobile

Avantages

- »Écrivez moins, faites plus »Cadre d'interface utilisateur
- Compatible multiplateforme et multi-navigateur
- Compatible avec d'autres frameworks
- Facile à apprendre et à développer des applications
- Inclut ThemeRoller pour créer un thème personnalisé sans écrire la ligne de code
- Le code évolue automatiquement de l'écran du mobile à l'écran du bureau

JQuery Mobile

Désavantages



- Options limitées pour les thèmes CSS
- Plus lent sur les mobiles
- Prend du temps lorsqu'il est combiné avec d'autres frameworks mobiles
- Difficile de fournir une conception visuelle personnalisée complète
- Toutes les fonctionnalités de l'appareil ne sont pas accessibles

JQuery Mobile

Outils de développement



- Système d'exploitation : Windows, Linux, OS X
- Langages de programmation : HTML, CSS, JS
- Frameworks : jQuery Mobile
- IDE : Eclipse, Netbeans, IntelliJ, Sublime Text, etc.
- Divers : grande collection de démos et la suite de tests unitaires de la bibliothèque

- Framework HTML front-end open source construit sur Angular et Apache Cordova
- Fournit des outils et des services pour créer une interface utilisateur mobile avec une apparence native
- Besoin d'un wrapper natif pour pouvoir fonctionner sur des appareils mobiles
- Gère toutes les interactions d'apparence et d'interface utilisateur des applications mobiles
- Prend en charge une large gamme de composants mobiles natifs courants



- Architecture MVC angulaire : pour créer des applications riches d'une seule page optimisées pour les appareils mobiles
- Composants CSS : impliquaient presque tous les éléments dont une application mobile a besoin avec l'apparence native
- Composants JavaScript : couvrent tous les éléments mobiles qui ne peuvent être réalisés uniquement avec HTML et CSS
- Plugins Cordova : utilisés pour accéder aux fonctions natives de l'appareil avec du code JavaScript



- Utilitaire NodeJS : pour démarrer, créer, exécuter et émuler des applications Ionic
- Ionic View : pour télécharger, partager et tester des applications sur des appareils natifs
- Les composants peuvent être remplacés pour s'adapter aux conceptions des développeurs
- Application pré-générée avec des mises en page simples
- Les applications de build sont faciles à maintenir et à mettre à jour



- Système d'exploitation : Windows, Linux, OS X
- Langages de programmation : HTML, CSS, JS
- Frameworks : Ionic, NodeJS
- SDK : SDK Android
- IDE : tous les IDE HTML et JS et XCode
- XCode : si l'application est pour iOS



À venir



À venir



À venir



WebMobile

Avantages

- Application unique pour toutes les plateformes
- Utiliser n'importe quel langage/technologie Web
- Facile à construire et facile à entretenir
- Installer localement sur l'appareil et s'exécute en tant qu'application installée
- Ne nécessite pas de boutique d'applications
- Moins cher que natif et hybride



WebMobile

Désavantages

- Impossible d'interagir avec les utilitaires de certains appareils
- Plus lent que l'application native
- Moins interactif et moins intuitif
- Aucune icône sur le bureau
- Ne peut pas être soumis aux App Stores



WebMobile

Désavantages

- Doit être exécuté dans un navigateur
- Certains matériels et logiciels de l'appareil ne peuvent pas être utilisés
- Nécessite une connexion Internet pour fonctionner (HTML5 - résout cette limitation)

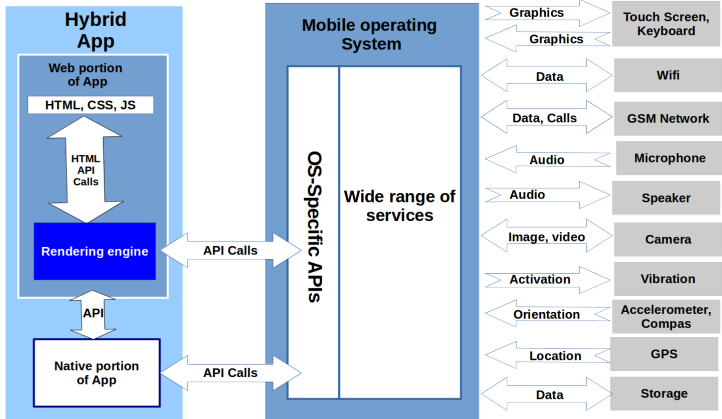
Hybrid Développement d'applications mobiles



- Exécuter dans un conteneur/WebView via un framework
- Application hébergée dans un composant WebView natif (Wrapper)
- WebView : composant d'interface utilisateur disponible sur le système d'exploitation mobile

Hybrid Développement d'applications mobiles

Interaction avec le téléphone





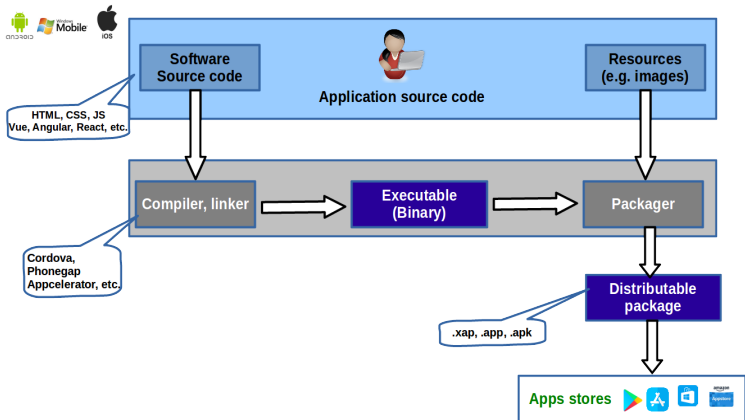
Hybrid Développement d'applications mobiles

Technologies clés

- Langages de programmation : HTML, CSS, JS
- Frameworks : jQuery mobile, Sencha Touch, Angular, ionic, Vue.JS, Cordova, PhoneGap, Appcelerator, Whitelight, etc.
- IDE : Eclipse, Netbeans, Sublime Text, Visual Studio Code, etc.

Hybrid Développement d'applications mobiles

Les technologies





Hybride Développement d'applications mobiles

Cordoue

- Cadre open source
- Utilisé pour créer des applications mobiles hybrides pouvant utiliser de nombreuses fonctionnalités de téléphones mobiles
- Offre un pont pour la connexion entre l'application Web et l'appareil mobile
- Les applications s'exécutent dans des wrappers ciblés sur chaque plateforme
- Offre une plateforme unique pour créer des applications mobiles hybrides : IOS, Android, Windows Phone, Amazon-fireos, blackberry, Firefox OS, etc.
- Gagnez du temps de développement par rapport au développement d'applications natives



Hybride Développement d'applications mobiles

Cordoue : Caractéristiques

- Cordova CLI pour :
 - Démarrer des projets
 - Création de processus pour différentes plateformes
 - Installation de plugins
- Composants de base Cordova : ensemble de composants de base dont chaque application mobile a besoin
- Plugins Cordova : API utilisée pour implémenter des fonctions mobiles natives



Hybrid Développement d'applications mobiles

Espace téléphonique

- Framework de développement mobile développé par Adobe System
- Produit des applications pour toutes les plates-formes de système d'exploitation mobiles populaires : iOS, Android, BlackBerry et Windows Mobile OS, etc.
- Permet aux utilisateurs de télécharger du contenu de données sur le site Web et de le convertir automatiquement en divers fichiers d'application
- Les candidatures sont téléchargées sur <https://build.phonegap.com/> et l'application mobile est générée automatiquement



Hybride Développement d'applications mobiles

Cordova et PhoneGap : outils de développement

- Système d'exploitation : Windows, Linux, OS X
- Langages de programmation : HTML, CSS et JS
- Frameworks : framework Cordova, NodeJS
- IDE : XCode, tous les IDE HTML et CSS



Hybride Développement d'applications mobiles

Avantages désavantages

Avantages :

- Multi-plateforme : application unique pour toutes les plateformes
- Rapide à développer et à mettre à jour
- Accès à de nombreux utilitaires d'appareils à l'aide d'API
- Aucun navigateur requis
- Beaucoup moins cher qu'une application native

Désavantages:

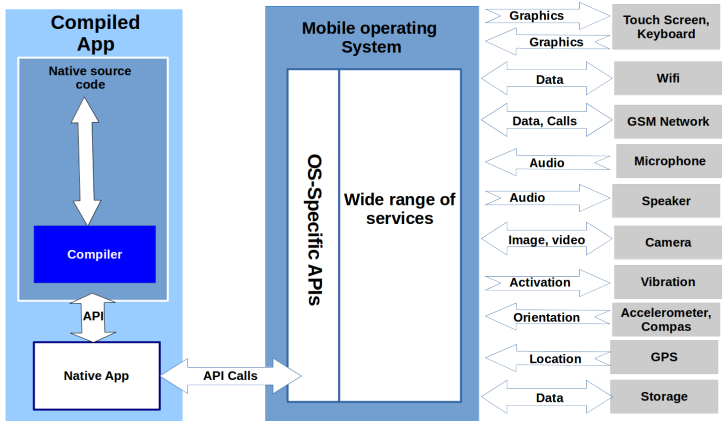
- Plus lent que les applications natives
- Moins interactif que les applications natives
- Plus cher que les applications Web



- Des applications essentiellement natives
- Utilise les technologies Web, mais convertit les applications en composants natifs
- Proche des applications natives en termes de performances

Compilé le développement d'applications mobiles

Interaction avec le téléphone



Ccompilé

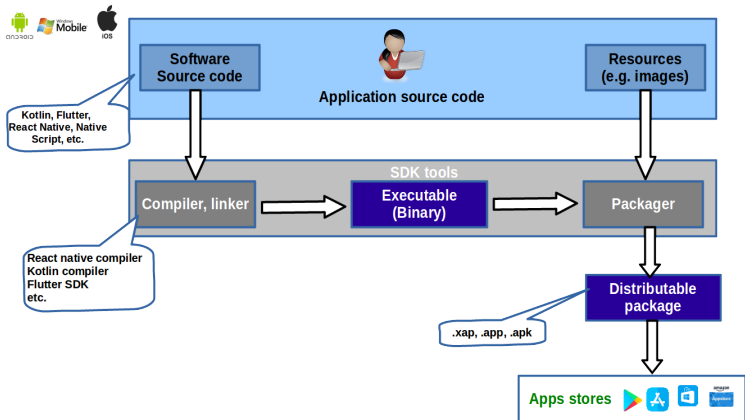
Technologies clés



- Langages de programmation : HTML, CSS, JS, Dart
- Frameworks : React Native, Kotlin Native, Native Script, Flutter, etc.

Compilé le développement d'applications mobiles

Les technologies





Compilé le développement d'applications mobiles

Kotlin

- Langage de programmation open source introduit par JetBrains
- Influencé par d'autres langages de programmation : Java, Scala, Groovy, etc.
- Combine des fonctionnalités orientées objet et fonctionnelles dans une plateforme unique
- Défini comme langage de programmation officiel en 2017 par Google



Compilé le développement d'applications mobiles

Kotlin : avantages et inconvénients

Avantages :

- Facile à apprendre
- Meilleures performances et faible autonomie

Désavantages:

- Déclaration d'espace de noms : les fonctions doivent être déclarées au niveau supérieur
- Pas de déclaration statique : peut poser problème au développeur Java conventionnel

Compilé le développement d'applications mobiles

Kotlin : outils de développement



- Système d'exploitation : Windows, Linux, OS X
- SDK : JDK 8 ou supérieur
- IDE : Eclipse, Netbeans, IntelliJ IDEA
- Plugin Kotlin pour Eclipse



Compilé le développement d'applications mobiles

Réagir natif

- Framework JS pour créer des applications mobiles natives Android et iOS
- Conçu sur la tête de React par Facebook
- Livré avec des cycles de développement rapides et des composants d'interface utilisateur déclaratifs et autonomes
- Offre une grande quantité de composants et d'API intégrés
- Les développeurs peuvent utiliser des composants natifs contrôlés par JS
- Les applications développées ne peuvent pas être distinguées des applications créées à l'aide d'Objective-C ou de Java.



Compilé le développement d'applications mobiles

React Native : outils de développement

- Système d'exploitation : Windows, Linux, OS X
- Langages de programmation : HTML, CSS, JS, Python
- Frameworks : NodeJS, create-react-native-app
- SDK : Python, JDK 8 ou supérieur, fil
- IDE : Android Studio, XCode
- Divers:
 - Réagir à la CLI native
 - »expo » : client dans l'appareil Android et scannez le code QR
 - fil, npm

Compilé le développement d'applications mobiles

NatifScript



- Framework open source développé par Progress Telerik
- Utilisé pour créer des applications natives iOS et Android



Compilé le développement d'applications mobiles

NativeScript : outils de développement

- Langages de programmation : HTML, CSS, JS, XML
- Frameworks : Angular, TypeScript, JS, NodeJS
- SDK : SDK Android, JDK 8 ou supérieur
- IDE : Playground qui est un WebIDE, XCode
- Divers : médecin tns pour vérifier la dépendance et résumer les résultats



Compi dirigé le développement d'applications mobiles

Avantages désavantages

Avantages :

- Les développeurs peuvent utiliser les connaissances JS existantes pour créer des applications mobiles natives
- Le code peut être partagé entre les plateformes

Inconvénients : nécessité d'écrire du code spécifique à la plate-forme pour les nouveaux composants natifs → tâche fastidieuse

Compil

dirigé le développement d'applications mobiles



- Écrivez un code et compilez-le sur des applications mobiles de différentes plates-formes
- Les applications obtenues après la compilation ne s'exécutent pas dans une vue Web

Compi dirigé le développement d'applications mobiles

Battement



- Framework open source pour créer des applications mobiles de haute qualité et hautes performances
- Permet de développer des applications pour Android et iOS
- Framework moderne et réactif
- Fournit un SDK pour écrire une application mobile dans Dart
- Dart : langage de programmation développé par Google



Compi dirigé le développement d'applications mobiles

Flutter : avantages

- Simple, puissant, efficace et facile à comprendre
- Exécute la même interface utilisateur pour plusieurs plates-formes
- Moins de tests : tests pour les deux plates-formes
- Application haute performance
- Des widgets magnifiques et personnalisables
- Développement rapide et excellents outils de développement

Compi dirigé le développement d'applications mobiles

Flutter : inconvénients



- Les développeurs doivent apprendre un nouveau langage
- L'interface utilisateur et la logique sont mélangées : un codage intelligent peut être utilisé pour les séparer



Compilé le développement d'applications mobiles

Flutter : outils de développement

- Système d'exploitation : Windows, Linux, OS X
- SDK : SDK Flutter, SDK Android
- IDE : Eclipse avec plugins Flutter et Dart, studio Android / XCode
- Un émulateur Android/simulateur iOS ou un véritable appareil Android
- Divers : Flutter doctor pour vérifier si toutes les exigences du développement du flutter sont remplies

Difficultés lors du développement d'applications mobiles

- Séparation entre les équipes opérationnelles et de développement
- Manque de collaboration entre l'équipe de développement
- Manque de communication entre l'équipe de développement

Solution : Briser le mur entre le développement et l'exploitation

Mobilet DevOps

Présentation du DevOps



- Approche de développement logiciel
- Se concentre sur la collaboration efficace entre les développeurs de logiciels, le personnel opérationnel et les chefs de projet
- Permet de surmonter les défis liés à la livraison continue de logiciels
- Utilisé pour rendre l'intégration continue et la livraison continue plus efficaces
- Les équipes Dev et Ops se sont réunies pour relever les défis de la livraison continue d'applications

Mobile DevOps

Présentation du DevOps



- Facilitez la livraison transparente et rapide des applications mobiles, de la conception à la production finale, en passant par la livraison et le déploiement dans l'App Store.
- Principe clé :
 - Intégration continue:
 - Livraison continue : déployer le logiciel et l'environnement sur lequel il s'exécute, automatiquement et à la demande, à n'importe quelle étape du cycle de développement



- Livraison continue d'applications mobiles
- Permet aux clients de réduire les délais de mise sur le marché et les nouvelles fonctionnalités
- Saisir facilement de nouvelles opportunités de marché
- Défis:
 - Différents fournisseurs de systèmes d'exploitation
 - Différentes versions du système d'exploitation
 - Différents fabricants de smartphones



Mobile DevOps

Adoption

- Planification continue : réunir l'équipe (développeurs, managers, opérations, etc.) sur la même plateforme pour finaliser le périmètre de l'application mobile dans la préparation d'un plan de lancement
- Intégration continue:
 - L'utilisation du système de contrôle du code source
 - Le code soumis par un développeur fonctionne avec le code soumis par les autres
 - Construit fréquemment intégré au code développé
- Tests continus :
 - Impossible de couvrir le grand jeu des appareils mobiles (fabricants, OS) avec des tests manuels
 - — → Accueillir les tests automatiques

Mobile DevOps

Adoption



- Surveillance continue : effectuée via des SDK tiers
- Livraison continue : déployez chaque modification dans un environnement de type production
- Déploiement continu :
 - Prochaine étape de la livraison continue
 - Chaque modification approuvée par les tests continus est automatiquement déployée dans l'environnement de production.

Mobile DevOps

Pourquoi?



- Le mobile nécessite des itérations rapides
- Automatisation du déploiement :
 - Réduit le temps nécessaire au déploiement de nouvelles itérations
 - Permet aux équipes de réagir plus rapidement aux commentaires des utilisateurs
- Les applications mobiles sont moins indulgentes : les utilisateurs sont beaucoup moins indulgents lorsqu'il s'agit d'une expérience utilisateur moins bonne
- Les outils mobiles évoluent constamment : de nouvelles technologies pour répondre aux besoins toujours croissants du marché

Mobile DevOps

Pourquoi?



- Mobile DevOps permet :
 - Rapports et commentaires instantanés
 - Les développeurs pour résoudre les erreurs et corriger les bugs plus rapidement
- Certaines fonctionnalités peuvent ne pas fonctionner comme prévu sur toutes les plateformes prises en charge
- Les applications fonctionnent bien lors des tests mais échouent entre les mains de l'utilisateur est un scénario courant

Mobile DevOps

Outils



- git, svn, Github, GitLab, etc. : Gestion du code source
- Solution intégrée de cycle de vie de développement mobile, etc. :
Visual Studio App Center, Azure DevOps
- Jenkins : intégration continue avec
- pCloudy, Appium, Appcelerator, etc. : tests en continu



Mobile Devops

Comment inclure DevOps dans votre projet d'application mobile ?

1. Agile Planning : déterminer le périmètre de l'application mobile et planifier sa sortie
 2. Intégration constante : suivez les meilleures normes et pratiques de contrôle de version
 3. Tests appropriés : assurez-vous que le produit livré au client est de bonne qualité
 4. Surveillance précise :
 - L'application mobile fonctionne bien conformément aux exigences
 - L'application mobile est stable dans l'environnement de production
 - Livraison plus rapide : assurez-vous que le code source est prêt à être déployé dans un environnement de production
- 74 / 84 Rendre la version stable disponible pour le déploiement dès que possible



Mobile Devops

Comment inclure DevOps dans votre projet d'application mobile : résumé

1.Intégration et livraison continues :

- Le code doit être facilement intégré aux autres
- Les actifs de développement d'applications (scripts, fichiers texte, configurations, documents, code, etc.) sont traçables

2.Tests et surveillance :

- Effectuez des tests dans un environnement réel en tenant compte de la taille de l'écran, de la résolution, des capacités, des fonctionnalités, etc.
- Testez l'application de manière continue
- Les tests automatisés des applications nous permettent d'améliorer le processus de test et d'identifier les bugs très rapidement

3.Contrôle de qualité:

- Surveillez en permanence les notes et les commentaires sur l'App Store mobile
- Éliminez les bugs et ouvrez la voie au changement et à la croissance futurs

Mobile DevOps



Avantages de DevOps dans le développement d'applications mobiles

- Plus de temps pour l'innovation
- Une meilleure collaboration
- Délai de livraison rapide
- Environnement stable et sécurisé pour des déploiements faciles
- Corrections de bugs rapides
- Automatiser les tâches répétitives sans trop se soucier des erreurs
- etc.

Chousadopter une démarche de Développement Mobile



- Grande variété de types de systèmes d'exploitation et d'appareils, tous évoluant à des rythmes différents
- Grande variété de frameworks et d'outils de développement pour chaque approche de développement mobile
- Le succès dépend de la nécessité de fournir rapidement des applications offrant la qualité et les fonctionnalités exigées par les utilisateurs, sur une large gamme de tailles d'écran, de formats de présentation et de capacités d'appareils.
- La prolifération rapide des téléphones intelligents et la sortie de nouvelles tablettes ont rendu le choix de la bonne solution de mobilité plus complexe que jamais.

Chousdopter une démarche de Développement Mobile

Cas d'utilisation 1



Les principaux utilisateurs sont des médecins et autres personnels de santé, habitués à vivre une expérience riche sur leur appareil mobile. Les fonctionnalités impliquaient plusieurs canaux de communication, notamment le chat et les appels téléphoniques VoIP.

Un bon choix devrait être Android natif ou iOS

Chousadopter une démarche de Développement Mobile

Cas d'utilisation 2



L'application existante permet aux utilisateurs de faire des achats en ligne à l'aide de leurs appareils mobiles. Le client souhaite ajouter un nouveau canal pour ses consommateurs. Le plus grand nombre de consommateurs doit être atteint tout en fournissant des services sur plusieurs types d'appareils.

Solution Web mobile



Chousadopter une démarche de Développement Mobile

Cas d'utilisation 3

L'objectif de l'entreprise est de construire un système qui fonctionnera sur plusieurs smartphones (iPhone, appareils Android et BlackBerry). Les principales fonctionnalités sont les suivantes : récupérer des informations sur les prix, saisir des éléments de campagne sur un devis, obtenir les données GPS sur la localisation du client. Bien qu'elle ne soit pas familière avec les technologies natives, l'équipe informatique était expérimentée avec JS et CSS, qui sont deux technologies de base pour PhoneGap.

Application mobile multiplateforme avec PhoneGap



Why développer des applications pour smartphone ?

- Accès Internet partout
- Réseaux sociaux
- Millions d'utilisateurs
- Norme ouverte
- Changer la façon dont les gens travaillent, font leurs achats, s'informent, etc.
- De nombreux cas d'utilisation :
 - Vendre des produits et services
 - Augmenter de manière productive et accroître la sensibilisation
 - Accéder à l'information
 - etc.

Kyeux q Questions sur le choix des technologies

- Qui utilisera l'application ?
 - Identification de l'audience : permet de connaître les plateformes à supporter et le type d'expérience utilisateur à proposer
 - Les exigences en matière de plate-forme sont l'élément clé de la sélection de la technologie
- Quel type d'expérience les utilisateurs attendent-ils ?
- Quelle fonctionnalité est requise ?
 - Une quantité plus élevée d'exigences d'information
 - Un montant plus élevé d'exigences transactionnelles
 - Exigences spécifiques à l'appareil
 - L'application nécessite un accès hors ligne à des fichiers volumineux tels que des catalogues PDF



Kyeux q questions

- Quelle est l'importance de la compatibilité multiplateforme ?
- L'application doit-elle s'exécuter hors ligne ou lorsque la connexion est mauvaise ?
- Existe-t-il des exigences spécifiques à l'industrie ?
- De combien de temps disposez-vous pour créer l'application ?
- Votre personnel possède-t-il les compétences nécessaires ?
- Quel est le coût total de possession ?
- Dans combien de temps avez-vous besoin de nouvelles fonctionnalités d'appareil ou de système d'exploitation ?



Je tiens à exprimer ma gratitude particulière à tous les étudiants de Master 1 de l'Université de Yaoundé 1 qui ont contribué à l'amélioration de cet accompagnement. Spécialement:

-