# Introduction

Il ne faut jamais oublier que l’informatique est un élément essentiel et nécessaire dans la vie moderne, où l’ère actuelle a connu une révolution énorme dans le développement des logiciels et des méthodes de conception, en plus d’élargir le champ de son utilisation et les plateformes sur lesquelles elle opère, comme les applications et les mini programmes qui fonctionnent à l’aide d’un outil appelé Smartphone.

Dans ce petit chapitre, on va faire une petite présentation des technologies des Smartphones et applications mobiles, c’est ce qui va nous aider à se familiariser avec les différents composants du projet.

# C’est quoi un Smartphone ?

Un Smartphone désigne un téléphone portable intelligent doté de fonctionnalités diverses évoluées qui s'apparentent à celles d'un ordinateur, il peut exécuter des applications grâce à un système d'exploitation spécialement conçu pour mobiles « OS Mobile », et donc en particulier fournir des fonctionnalités en plus de celles des téléphones mobiles classiques comme : l'agenda, la télévision, le calendrier, la navigation sur le Web, la consultation et l'envoi de courrier électronique, etc.

# C’est quoi un OS mobile ?

Un OS Mobile est un système d’exploitation mobile qui peut être défini comme le logiciel permettant à un Smartphone ou un téléphone mobile basique de fonctionner. Ce type d’OS se concentre entre autres sur la gestion de la connectivité sans fil et celle des différents types d'interface.

# OS Mobiles Utilisés par les Smartphones

## Androïde OS

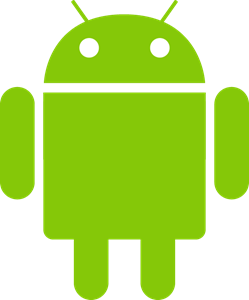


Figure 1 : Logo d'Androïde

Androïde OS est basé sur le noyau Linux et développé actuellement par Google. Androïde a été développé par l’Open Handset Aliance. Il a été annoncé en 2007 et il est devenu une plateforme ouverte en 2008. Le système avait d'abord été conçu pour les Smartphones et tablettes tactiles, puis s'est diversifié dans les objets connectés et ordinateurs comme les télévisions, les voitures, les ordinateurs et les montres intelligentes.

### Avantage

* Variété de téléphones disponible : Androïde OS n’est pas réservé à une seule marque de téléphone, il est donc possible pour une entreprise d’équiper ses employés en fonction de leurs besoins et utilisations.
* Le prix : La multitude de Smartphones disponibles permet logiquement à l’entreprise de maitriser ses coûts et d’adapter le choix de l’équipement de ses employés en fonction de son budget.
* Androïde OS dispose aussi d’un très grand nombre d’applications sur son « Play Store ».

### Inconvénient

* La complexité de l’OS : Androïde OS est riche et complexe, parfois difficile à appréhender dans sa globalité par les utilisateurs.
* Permissivité du système : Androïde OS est assez permissif, pouvant entraîner des failles dans la sécurité.

## IOS



Figure 2 : Logo d'IOS

IOS, anciennement iPhoneOS, est développé par Apple pour plusieurs de ses appareils. Il est dérivé de MacOs dont il partage les fondations (le noyau hybride XNU basé sur le micronoyau Mach, les services Unix et Cocoa, etc.). IOS comporte quatre couches d'abstraction, similaires à celles de MacOs : une couche « Core OS », une couche « Core Services », une couche « Media » et une couche « Cocoa ». Le système d'exploitation occupe au maximum 3 Go de la capacité mémoire totale de l'appareil, selon l'appareil.

### Avantage

* La facilité d’utilisation de l’OS : Apple a créé avec succès un système d’exploitation des plus simples et intuitif à utiliser, ce qui a largement joué en faveur de la marque pour la diffusion de ses produits en entreprise.
* Le parc applicatif : IOS recense un très grand nombre d’applications, dont beaucoup d’entre elles directement applicables au monde de l’entreprise.
* Le positionnement haut-de-gamme de l’iPhone : Perçu comme un objet statutaire, l’iPhone est souvent le Smartphone de prédilection des cadres / managers / commerciaux d’une entreprise.
* Le niveau de sécurité élevé.

### Inconvénient

* Les Smartphones Apple sont des produits chers, et la marge de négociation avec Apple reste des plus faibles.
* Les fréquentes mises à jour d’IOS peuvent être un frein pour les DSI recherchant la plus grande uniformité et la limite des coûts dans l’entreprise.
* La mémoire disponible sur le téléphone. Il n’est pas possible de rajouter de cartes mémoire sur un iPhone pour en augmenter la capacité de stockage. Il faut donc choisir à l’achat du téléphone le nombre de gigas disponible et s’y tenir par la suite.

## Windows Mobile/Phone OS

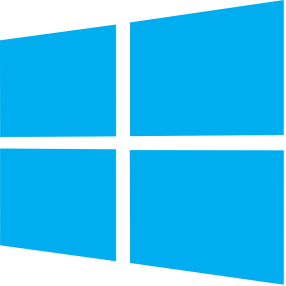


Figure 3 : Logo de Windows

Windows Phone OS est développé par Microsoft pour succéder à Windows Mobile, sa précédente plateforme logicielle qui a été renommée pour l'occasion en Windows Phone Classique. Contrairement au système qu'il a remplacé, Windows Phone 7 était d'abord principalement destiné au grand public. Cependant à partir de Windows Phone 8, Microsoft a proposé des fonctions avancées pour les entreprises ainsi qu'un espace d'applications réservé aux professionnels. À partir de novembre 2015, Windows Phone disparaît progressivement et est remplacé par la version mobile de Windows 10.

### Avantage

* L’interface fluide et intuitive de l’OS. Windows Phone propose une interface simple à prendre en main, pratique pour les utilisateurs novices. De plus, il existe une réelle continuité entre l’OS Windows Phone et Windows (PC), très largement répandu en entreprise.
* Les Smartphones Windows sont tous équipés d’une version Office portable, permettant aux utilisateurs de lire, écrire, ou modifier un document Word, Excel ou Powerpoint. Un argument non négligeable dans le monde de l’entreprise.
* Le prix des téléphones. Il est facile de s’équiper en Windows Phone tout en limitant les coûts. De plus, il est possible pour un DSI, ou un responsable des achats, enclin à la négociation de profiter d’une certaine marge de manœuvre pour négocier ses prix.

### Inconvénient

* Les applications disponibles. Avec près de 300 000 applications disponibles, le store de Windows reste loin derrière IOS et Androïde qui ont tous les deux environ 1 million d’applications disponibles.
* Incertitude concernant la viabilité de l’OS. Les résultats, en termes de part de marché, de Windows Phone peuvent laisser craindre que l’OS ne survive pas longtemps à la domination d’Androïde et IOS.

## BlackBerry OS



Figure 4 : Logo de BlackBerry

BlackBerry OS est conçu par la société canadienne Research In Motion (RIM), maintenant connue sous le nom de BlackBerry. Il s'agit d'un système multitâche. Le système est surtout connu pour son support natif des courriels à travers le protocole Mobile Information Device Profile (MIDP 1.0), et plus récemment un sous-ensemble de MIDP 2.0 qui permet une synchronisation complète avec les messageries d'entreprise telles que Microsoft Exchange ou IBM Lotus Domino.

### Avantage

* La sécurité des données. BlackBerry est particulièrement reconnu pour la sécurité de l’architecture de ses produits. Aux dernières nouvelles, le ministère de la défense américain, ainsi que certaines banques, seraient toujours clients BlackBerry.
* BlackBerry est centré sur le monde de l’entreprise. BlackBerry a développé des solutions spécialement adaptées à l’entreprise : par exemple BES 12, une solution EMM multiplateforme conçue pour gérer en toute sécurité les téléphones des principales plateformes mobiles d’entreprise. Autre exemple, BlackBerry Blend,  une solution pour simplifier et sécuriser l’accès des employés à leurs données personnelles ou professionnelles à partir de leur Smartphone.

### Inconvénient

* Les applications disponibles sur le store. De plus en plus d’éditeurs d’applications font l’impasse du développement de leurs applications sur BlackBerry, ce qui limite grandement le nombre d’applications disponibles. Afin de réduire cet inconvénient, il est désormais possible d’installer des applications pour Androïde sur un Smartphone BlackBerry.
* Peu de choix de Smartphones. L’utilisation de ce système d’exploitation se limite aux Smartphones de la marque BlackBerry, ce qui représente une gamme assez peu variée de modèles.
* Incertitude concernant la viabilité de l’OS. De même que pour Windows Phone, les résultats de BlackBerry, laissent craindre sur la pérennité de l’OS.

# Applications Mobiles



Figure 5 : Applications Mobiles

Une application mobile est un logiciel applicatif développé pour un appareil électronique mobile, tel qu'un téléphone portable, un Smartphone, une tablette tactile, ou encore certains ordinateurs fonctionnant avec le système d'exploitation Windows Phone. Elles sont pour la plupart distribuées depuis des plateformes de téléchargement telles que l'App Store (plateforme d'Apple), le Google Play (plateforme de Google / Androïde), ou encore le Windows Phone Store (plateforme de Microsoft).

## Développement des applications mobiles

Chacun des OS mobile dispose d'un langage de programmation propre à lui, permettant le développement de ses applications :

* Les applications pour les terminaux Apple sont développées dans un langage principalement dédié à ces applications mobiles, le Swift.
* Celles pour Windows Mobile, sont développées en C#, langage aussi utilisé pour les programmes exécutables.
* Le système Androïde utilise, quant à lui, un langage universel, le Java, langage pouvant être utilisé pour les ordinateurs, le développement Web.

## Types d’applications mobiles

### Applications Natives

Une application native est une application mobile qui est développée spécifiquement pour un des systèmes d’exploitation utilisé par les Smartphones et tablettes (iOS, Androïde, etc.).  
Le fait de développer une application native permet généralement d’utiliser toutes les fonctionnalités liées au système d’exploitation visé (GPS, accéléromètre, appareil photo, etc.).

### Application Hybride

Une application hybride est une application utilisant le navigateur web intégré du support (Smartphone ou tablette) et les technologies Web (HTML, CSS et JavaScript) pour fonctionner sur différents OS (IOS, Androïde, Windows Phone, etc.). Une telle application utilise les fonctionnalités natives des Smartphones et peut être distribuée sur les plateformes d’applications telles que l'App Store, le Google Play, etc.

# Androïde

## Pourquoi Androïde ?

## Versions Androïde

Depuis l’apparition d’Androïde pour la première, Google a développé plusieurs versions où chacune de ces versions avait des nouvelles fonctionnalités et caractéristiques, mais avec la révolution de la technologie mobile, l’utilisation des anciennes versions Androïde (de 1.0 jusqu’à 4.0) est devenu insuffisante pour suivre l’avancement courant. Voici un petit chronogramme illustrant toutes les versions depuis l’annoncèrent de la plateforme :



Figure 6 : Chronogramme des versions d'Androïde avec logo

## Architecture Androïde

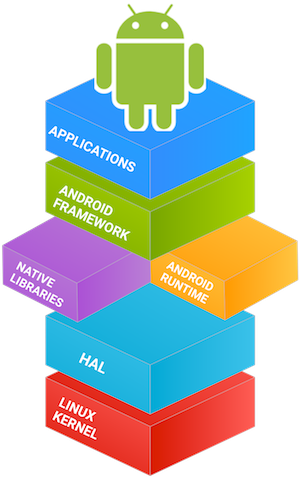


Figure 7 : Architecture d'Androïde

### Système

Le système Androïde est composé de plusieurs couches, la première couche est : Kernel linux 2.6.xx, au-dessus du Kernel il y a HAL "hardware abstraction layer" qui permet de séparer la plateforme logique du matériel, Au-dessus de cette couche d'abstraction on retrouve les bibliothèques C/C++ utilisées par un certain nombre de composants du système Androïde.  
Au-dessus des bibliothèques on retrouve l'Androïde Runtime, cette couche contient les librairies cœurs du Framework ainsi que la machine virtuelle exécutant les applications.  
Au-dessus la couche "Androïde Runtime" et des librairies cœurs on retrouve le Framework permettant au développeur de créer des applications. Enfin au-dessus du Framework il y a les applications.

#### Noyau Linux (Linux Kernel)

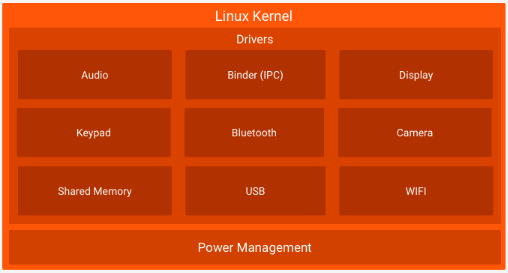


Figure : Noyau Linux

Androïde est basé sur un Kernel linux 2.6 qui est une version conçue spécialement pour l'environnement mobile, avec une gestion avancée de la batterie et une gestion particulière de la mémoire et tous les fonctions fondamentales du système d'exploitation telles que la gestion des processus, des fichiers, des entrées-sorties principales, et des fonctionnalités de communication, mais Il ne possède pas de système de fenêtrage natif. C'est cette couche qui fait en sorte qu'Androïde soit compatible avec tant de supports différents.

#### Couche d'abstraction matérielle (HAL)

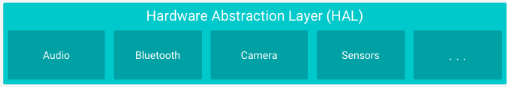


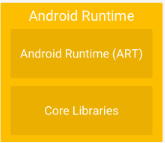
Figure : Couche d'abstraction matérielle

Est une spécification et un utilitaire logiciel qui traque les périphériques du système informatique, elle fournit les interfaces que doivent implémenter les drivers Kernel et sépare la plateforme logique des interfaces matérielles, son but est de faciliter le portage des librairies sur différents matériels, et éviter aux développeurs d'implémenter manuellement le code spécifique à un périphérique. Cette couche implémente un certain nombre de fonctions spécifiques au matériel : interfaces d'entrées-sorties, contrôleur d'interruptions, caches matériels, mécanismes de communication multiprocesseur...etc.

#### Bibliothèques C / C ++ natives

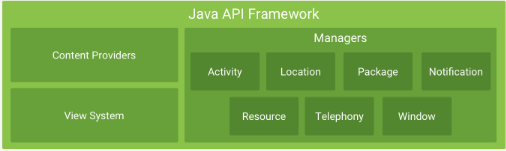


#### Android Runtime (ART)



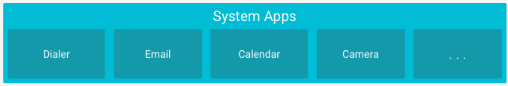
ART est un environnement d'exécution utilisé principalement par le système d'exploitation, cette couche se situe au-dessus des bibliothèques C/C++, elle se compose du cœur du Framework et de la machine virtuel dalvik. Il utilise la compilation anticipée (une compilation qui traduit un langage évolué en langage machine avant l'exécution d’un programme), en compilant l'application à son installation, sans besoin ultérieur d'interprétation, il permet ainsi d'augmenter les performances, donc d’augmenter la durée de vie de batterie. Les applications Java développées pour Androïde doivent être compilées au format dalvik exécutable (.dex) avec l'outil dx.

#### Java API Framework



L'ensemble complet des fonctionnalités du système d'exploitation Androïde est disponible à travers des API écrites en langage Java. Ces API constituent les blocs de construction dont vous avez besoin pour créer des applications Androïde en simplifiant la réutilisation des composants et services de base du système modulaire. Le Framework est situé au-dessus de l'Androïde Runtime et des librairies et Il fournit des API permettant aux développeurs de créer des applications riches.

#### Application système



### Application

Une application mobile est composée d’éléments suivants :

##### Les composants principaux

* Fragments : Représente une partie de l’interface utilisateur dans une activité.
* Views : Les éléments UI dessinés à l’écran (les boutons, les champs de texte, les listes des formulaires…etc).
* Layouts : Les hiérarchies qui contrôlent le format d'écran et l'apparence des vues (views).
* Intents : Petits objets qu'une activité peut transmettre au système d'exploitation Androïde, pour indiquer au système d'exploitation qu'une autre action ou activité est requise.
* Ressource : Les ressources dans Androïde font référence à des éléments externes tels que des images, des chaînes et d’autres éléments utilisés par l’application.
* Manifeste : Contient des informations sur votre package, y compris des composants de l'application tels que des activités, des services, des récepteurs de diffusion, des fournisseurs de contenu, etc.
* Activités (Activity) : Une activité représente la fenêtre qui sera affiché à l'utilisateur, de cette façon, une activité Androïde est très similaire à Windows dans une application de bureau, une application Androïde peut contenir une ou plusieurs activité (ça veut dire un ou plusieurs écran). Elle permet de gérer des fonctionnalités telles que l'appui sur la touche (MENU) ou l'affichage de messages d'alerte (Toast).

##### Les composants supplémentaires

* Les services : Un service est un composant qui s’exécute en arrière-plan et qui n’a pas une interface graphique. Il effectue des opérations à long terme tels qu’une lecture mp3 qui doit tourner en tâche de fond lorsque l’utilisateur se trouve dans une autre application.
* Récepteurs de diffusion (Broadcast receivers) : Un récepteur de diffusion est une application qui est à l'écoute des autres applications, elle répond aux messages diffusées provenant d’autres applications ou du système.
* Fournisseur de contenu (Content providers) : Les fournisseurs de contenu sont des gestionnaires de données, ils fournissent des données d’une application à d’autres sur demande, donc ils permettent le partage d’informations entre les applications.

# Les technologies de développement mobile (Framework)

## Android Studio :



Android Studio est un nouvel environnement pour développement et programmation entièrement intégré qui a été lancé le 8 décembre 2014 par Google pour les systèmes Androïde. Il a été conçu pour fournir un environnement de développement et une alternative à Eclipse qui est l'IDE le plus utilisé. Il permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Androïde. Il est basé sur IntelliJIDEA.

## Xamarin :



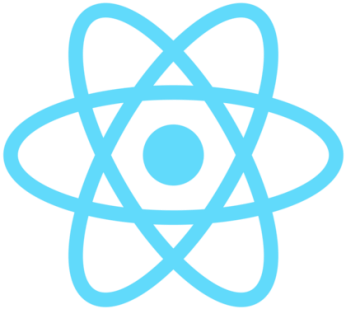
Xamarin est un Framework de développement mobile multiplateforme. Il permet tout simplement de développer son application sur IOS, Windows Phone et Androïde en même temps. Ce Framework est payant, il existe néanmoins des versions d'essais et il existe un programme pour étudiant afin d'obtenir une license indépendante gratuite. Il est principalement utilisé par les entreprises dans le but de réaliser des applications complexes. Ce Framework peut être utilisé au sein de Visual Studio, mais Xamarin possède aussi son propre IDE Xamarin Studio basé sur Mono Develop.

## Ionic :



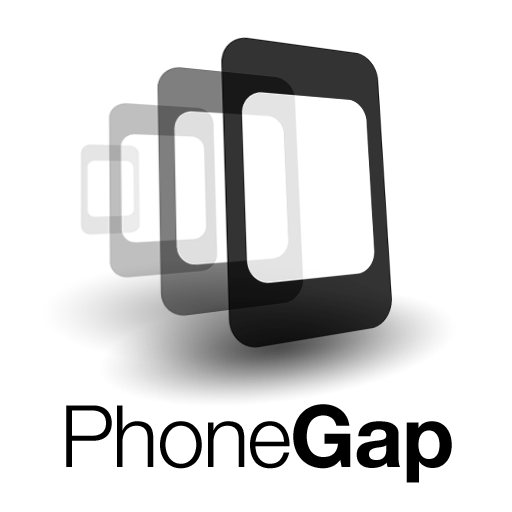
Ionic est un Framework open-source crée en 2013 par Max Lynch, Ben Sperry, et Adam Bradley. Deux versions distinctes sont disponibles, incompatibles entre elles : la première version, 1.3.3 se base sur AngularJS 1.5.3 tandis que la version 3.5.0 se base sur Angular 4.1.3 et TypeScript.

## React Native :



React Native est un Framework mobile hybride développé par Facebook depuis début 2015, basée sur React. Il continue d'évoluer avec le soutien de nombreux contributeurs sur Github. Le but de React Native est de pouvoir réutiliser le maximum de code entre les différentes plateformes (iOS et Android). Il offre un gain de temps considérable par rapport à du développement spécifique, tout en étant aussi performant.

## PhoneGap :



PhoneGap est un Framework open source pour créer des applis mobiles multiplateforme avec les technologies traditionnelles du web : HTML, CSS, JavaScript.

# Conclusion