

**ENSEIGNEMENT SUPERIEUR, UNIVERSITAIRE,
RECHERCHE SCIENTIFIQUE ET INNOVATIONS**

**INSTITUT SUPERIEUR DES SCIENCES
INFORMATIQUES ET GESTION DE BUNIA**

ISSIGE-BUNIA

B.P. 372 BUNIA



SECTION : SCIENCES INFORMATIQUES

Travail pratique Informatique mobile

Réalisé par : MWIRA WAYESU Daniel

NAIM USAMA

NATHANAEL MULUNDA Daniel

ANNEE ACADEMIQUE 2025-2026

Question 1 :

Explique l'évolution de l'informatique mobile depuis les années 1970 jusqu'à aujourd'hui

L'évolution de l'informatique mobile est le résultat de plusieurs décennies d'innovations technologiques qui ont transformé des appareils lourds et limités en outils intelligents et omniprésents dans notre quotidien.

Dans les années 1970 à 1980, l'informatique mobile en était encore à ses débuts. À cette époque apparaissent les premiers téléphones mobiles analogiques comme le Motorola DynaTAC. Ces appareils étaient très volumineux, coûteux et avaient une autonomie très limitée. Ils étaient principalement utilisés pour les appels vocaux et réservés à une minorité de personnes en raison de leur prix élevé.

Au cours des années 1990, une évolution importante se produit avec l'introduction des appareils numériques. Les assistants personnels numériques (PDA), tels que le Palm Pilot, permettent aux utilisateurs de gérer leurs contacts, leurs agendas et leurs notes. Parallèlement, la communication évolue avec l'apparition des SMS, qui rendent possible l'échange de messages écrits courts entre utilisateurs.

Au début des années 2000, l'informatique mobile entre dans une nouvelle phase avec l'apparition des premiers smartphones, notamment le IBM Simon, considéré comme le premier téléphone intelligent. Ces appareils combinent les fonctions de téléphone et d'ordinateur. L'arrivée des réseaux 3G permet également une amélioration significative de la vitesse de connexion à Internet, facilitant la navigation web et l'utilisation de services en ligne.

L'année 2007 marque une véritable révolution avec le lancement de l'iPhone par Apple. Ce téléphone introduit un écran tactile moderne, une interface intuitive et surtout un écosystème d'applications mobiles. Cette innovation transforme profondément l'usage des téléphones mobiles en les rendant accessibles et multifonctionnels.

Dans les années 2010, le marché connaît une expansion rapide avec l'arrivée du système Android développé par Google. Les smartphones deviennent de plus en plus performants, accessibles et répandus dans le monde entier. Les

applications mobiles se multiplient et couvrent tous les domaines : communication, éducation, santé, commerce, divertissement, etc.

Aujourd'hui, l'informatique mobile est devenue omniprésente. Les smartphones sont complétés par d'autres objets connectés comme les montres intelligentes (wearables) et les dispositifs de l'Internet des objets (IoT). Les technologies modernes telles que la 5G, l'intelligence artificielle et la réalité augmentée permettent des usages encore plus avancés, comme les assistants virtuels, les applications immersives ou encore la télémédecine.

Question 2 :

Quelles sont les principales plateformes mobiles et leurs outils ?

Les plateformes mobiles sont des environnements logiciels qui permettent de développer et d'exécuter des applications sur des appareils mobiles. Chacune possède ses propres outils, langages et spécificités.

La plateforme Android, développée par Google, est la plus utilisée au monde. Elle est open-source, ce qui signifie qu'elle peut être modifiée et adaptée par différents fabricants. Le développement d'applications Android se fait principalement avec les langages Java et Kotlin, et l'environnement de développement le plus utilisé est Android Studio. Cette plateforme est appréciée pour sa flexibilité et sa large communauté de développeurs.

La plateforme iOS, développée par Apple, est utilisée exclusivement sur les appareils de la marque (iPhone, iPad). Elle est connue pour sa sécurité, sa performance et son optimisation. Les développeurs utilisent les langages Swift et Objective-C, ainsi que l'environnement de développement Xcode. iOS offre une expérience utilisateur fluide et uniforme.

La plateforme Windows Phone, développée par Microsoft, a été une alternative aux deux premières, mais elle est aujourd'hui peu utilisée. Le développement se faisait principalement en C# avec l'outil Visual Studio. Malgré certaines innovations, cette plateforme n'a pas réussi à s'imposer sur le marché.

Enfin, les technologies cross-platform (multi-plateformes) permettent de développer une seule application compatible avec plusieurs systèmes (Android et iOS). Par exemple, Flutter utilise le langage Dart, React Native utilise

JavaScript, et Xamarin utilise C#. Ces outils permettent de réduire le temps et le coût de développement tout en assurant une large compatibilité.

Question 3 :

Décris les étapes du développement d'une application mobile

Le développement d'une application mobile est un processus structuré qui comprend plusieurs étapes essentielles, allant de l'idée initiale jusqu'à la maintenance après le lancement.

La première étape est l'analyse des besoins. Elle consiste à comprendre le problème à résoudre, définir les objectifs de l'application, identifier les utilisateurs cibles et préciser les fonctionnalités attendues. Cette phase est cruciale car elle permet de poser les bases du projet.

Ensuite vient la conception (design). À ce stade, on réalise des maquettes (wireframes) pour visualiser l'application. On travaille également sur l'interface utilisateur (UI) et l'expérience utilisateur (UX) afin de rendre l'application intuitive, attractive et facile à utiliser.

La troisième étape est l'architecture technique. Il s'agit de choisir les technologies à utiliser (Android, iOS, Flutter...), de définir le modèle d'architecture (comme MVC ou MVVM) et de mettre en place les bases de données et les API nécessaires au fonctionnement de l'application.

Après cela, on passe à la phase de développement. Les développeurs écrivent le code, intègrent les différentes fonctionnalités et réalisent des tests unitaires pour vérifier que chaque partie fonctionne correctement.

La cinquième étape est le test et la validation. L'application est testée dans son ensemble pour détecter les erreurs (bugs), vérifier ses performances et s'assurer qu'elle fonctionne correctement sur différents appareils et systèmes.

Ensuite, on procède au déploiement. L'application est publiée sur les plateformes de distribution comme Google Play Store ou App Store. Cela nécessite la préparation des fichiers d'installation (APK pour Android et IPA pour iOS).

Enfin, la dernière étape est la maintenance. Après le lancement, il est important de corriger les bugs, mettre à jour l'application et ajouter de

nouvelles fonctionnalités en fonction des besoins des utilisateurs et des évolutions technologiques.

Correction de TD1 et TD2

CORRIGÉ DU TD 1 : FONDAMENTAUX DE L'INFORMATIQUE MOBILE

1. Quel est le système d'exploitation principal pour les smartphones Android ?
 - Réponse : c) Android OS
2. Quel langage est principalement utilisé pour le développement d'applications Android ?
 - Réponse : b) Kotlin (Depuis 2017, c'est le langage recommandé par Google devant Java).
3. Quel environnement de développement est dédié au développement d'applications iOS ?
 - Réponse : c) Xcode (Développé par Apple).
4. Laquelle de ces options est un framework multiplateforme ?
 - Réponse : c) Flutter (Permet de compiler pour Android et iOS).
5. Quel composant hardware permet d'effectuer la gestion de réseau 4G ou 5G sur un smartphone ?
 - Réponse : c) Modem
6. La fonction principale d'un processeur (CPU) dans un smartphone est :
 - Réponse : b) Gérer l'exécution des instructions
7. Quel type de mémoire est utilisé pour le stockage permanent dans un smartphone ?
 - Réponse : b) ROM / Flash
8. Quel fournisseur offre la plateforme de développement pour iOS ?
 - Réponse : c) Apple

9. Laquelle de ces méthodes permet d'obtenir une image d'un smartphone avec ses composants internes ?
- Réponse : b) Schéma matériel
10. Quel est le rôle principal du système d'exploitation dans un smartphone ?
- Réponse : a) Gérer le matériel et les applications
11. Quelle technologie permet de développer des applications avec un seul code pour iOS et Android ?
- Réponse : b) Frameworks multiplateformes
12. Android Studio est basé sur quel langage principal ?
- Réponse : b) Java/Kotlin
13. Quel composant matériel sert à capturer des images dans un smartphone ?
- Réponse : c) Caméra
14. Dans le développement mobile, que signifie "UI" ?
- Réponse : a) User Interface (Interface Utilisateur).
15. Quel protocole est principalement utilisé pour la synchronisation de données en temps réel ?
- Réponse : c) Firebase Realtime Database (ou WebSocket)
16. Quel type de mémoire est volatile ?
- Réponse : a) RAM (Les données sont perdues lors de l'extinction).
17. Laquelle de ces plateformes est majoritairement open source ?
- Réponse : b) Android (Via le projet AOSP).
18. Quel est le principal avantage du développement cross-platform ?
- Réponse : b) Code unique pour plusieurs OS
19. En sécurité mobile, que désigne l'acronyme "MFA" ?

- Réponse : a) Multi-Factor Authentication (Authentification Multi-Facteurs).
20. Quelle technologie permet d'intégrer des notifications push dans une application mobile ?
- Réponse : b) Firebase Cloud Messaging (FCM) / Apple Push Notification Service (APNS)

CORRECTION DU TD 2 : INFORMATIQUE MOBILE

Exercice 1 : Conception d'une application de gestion de tâches

a) Quelles classes principales pourrais-tu définir pour modéliser cette application en UML ?

Pour une application de gestion de tâches (To-Do List), les classes principales sont :

1. Utilisateur (User) : Pour gérer le profil (nom, email, mot de passe).
2. Tache (Task) : La classe centrale contenant les attributs (id, titre, description, date_echeance, priorite, statut_termine).
3. Categorie (Category) : Pour classer les tâches (Travail, Personnel, Loisirs).
4. Notification : Pour gérer les rappels liés aux tâches.

b) Quelles fonctionnalités clés seraient accessibles pour l'utilisateur ?

Les fonctionnalités clés (souvent appelées CRUD) sont :

- Création : Ajouter une nouvelle tâche avec une priorité.
- Consultation : Afficher la liste des tâches (filtrées par date ou catégorie).
- Modification : Mettre à jour le contenu ou changer le statut d'une tâche.
- Suppression : Retirer une tâche de la liste.
- Synchronisation : Sauvegarder les données sur un serveur distant pour y accéder depuis n'importe quel appareil.

Exercice 2 : Sécurité et authentification

a) Quelles deux méthodes d'authentification peuvent être combinées dans MFA ?

Le MFA (Multi-Factor Authentication) combine généralement :

1. Facteur de connaissance : Quelque chose que l'utilisateur *connaît* (ex: mot de passe, code PIN).
2. Facteur de possession ou d'inhérence : Quelque chose que l'utilisateur *possède* (ex: code reçu par SMS, token matériel) ou quelque chose qu'il *est* (ex: empreinte digitale, reconnaissance faciale).

b) Explique comment la MFA renforce la sécurité de l'application.

La MFA ajoute une couche de défense supplémentaire. Même si un pirate parvient à voler le mot de passe (via phishing ou fuite de données), il ne pourra pas accéder au compte car il n'aura pas le deuxième facteur (le téléphone de la victime ou son empreinte). Cela réduit drastiquement les risques d'usurpation d'identité.

Exercice 3 : Développement avec framework multiplateforme

a) Quels sont les avantages de choisir Flutter ?

- Code unique (Single Codebase) : On écrit le code une seule fois en Dart pour Android et iOS.
- Hot Reload : Les modifications de code sont visibles instantanément sur l'émulateur sans redémarrer l'application.
- Performance : Flutter compile en code machine natif, offrant une fluidité proche des applications natives.
- UI Riche : Une vaste bibliothèque de widgets personnalisables pour des interfaces modernes.

b) Quelles sont les étapes principales pour démarrer avec Flutter ?

1. Installation du SDK : Télécharger et configurer le SDK Flutter.
2. Configuration de l'IDE : Installer VS Code ou Android Studio avec les plugins Flutter/Dart.
3. Création du projet : Utiliser la commande `flutter create mon_projet`.

4. Développement : Écrire le code dans le fichier main.dart.
5. Test : Lancer l'application sur un émulateur ou un appareil physique avec flutter run.

Exercice 4 : Optimisation énergétique

a) Quelles mesures pouvez-vous prendre pour réduire la consommation de la batterie ?

- Réduction de la précision : Utiliser le fournisseur réseau (Wi-Fi/Antennes) au lieu du GPS pur quand une haute précision n'est pas requise.
- Augmentation de l'intervalle de mise à jour : Ne demander la position que toutes les 5 ou 10 minutes au lieu de chaque seconde.
- Géorepérage (Geofencing) : N'activer le GPS que lorsque l'utilisateur entre ou sort d'une zone spécifique.
- Batching : Envoyer les données de localisation au serveur par paquets au lieu d'une connexion constante.

b) Quelles fonctions ou API éviter ou limiter ?

- `PRIORITY_HIGH_ACCURACY` : À utiliser avec parcimonie car elle force l'utilisation du GPS très précis et gourmand.
- Mise à jour constante en arrière-plan : Éviter de garder un *WakeLock* actif qui empêche le processeur de passer en mode veille.
- Scan Wi-Fi/Bluetooth constant : Très énergivore s'il est combiné au GPS.