

# Développement d'applications avec un utilisateur certifié Swift

## Objectifs

- Installer l'environnement de développement
- Créer et structurer un projet
- Programmer en Swift
- Décrire et implémenter le design pattern View-Controller
- Créer des interfaces utilisateurs avec la librairie UIKit
- Mettre en oeuvre les patterns de navigation
- Gérer la persistance locale des données
- Échanger des données avec un serveur
- Intégrer les fonctionnalités de l'appareil (Capteurs, Partage, Message, etc)
- Optimiser, construire et déployer une application

## Prérequis

- Maîtriser un langage de programmation (C#, Java, C++, PHP, JavaScript, Python)

## Programme

### Présentation d'iOS

- Historique de la plate-forme
- Diffusion actuelle
- Architecture et aspects techniques

### Présentation de Swift

- Pourquoi le besoin d'un nouveau langage
- La syntaxe et les règles de Swift
- Swift dans iOS

### Environnement de développement

- Découverte de Xcode et du Simulateur
- Structure des projets iOS
- Framework Foundation
- Création de tests unitaires
  - - Atelier : Installation de l'environnement de développement, génération d'un projet. Présentation des outils de débogage et de la console

### Les éléments fondamentaux d'iOS

- Les composants applicatifs : ViewController, View, AppDelegate
- Présentation de la gestion de la mémoire, gérée ses références
- Les composants d'interactions : Les closures, notifications système et les delegates

- Le cycle de vie d'une vue, les différents états d'une application
  - - Atelier : Réalisation d'une application permettant de montrer les différents états d'une application et d'une vue.

#### Création d'interface utilisateur

- Présentation de UIKit (liste, collection ....)
- Créer une interface grâce au fichier XIB
- Hiérarchie entre les vues, gérer les vues dans le code
- Gérer des layouts différents pour iPhone et iPad

#### Le fonctionnement AutoLayout

- Gérer la rotation
  - - Atelier : Réalisation d'une application gérant la rotation et layouts différents sur iPhone/iPad

#### La communication entre composants applicatifs

- Utilisation des delegates/closures/notification et dans quelle condition
- Navigation entre les vues controllers
  - - Atelier : Ajouter des éléments dans une liste/accéder au détails d'un élément

#### Tâches asynchrones et tâches de fond

- Créer des tâches asynchrones : GCD, NSTimer
  - - Atelier : Gestion d'une barre de progression

#### Le stockage de données

- Gestion des préférences
- La gestion des fichiers sous iOS
- Présentation de CoreData
- Savoir que ses données ont changé
  - - Atelier : Implémenter un modèle de données sur iOS, intégrer ce modèle sur une liste

#### Communiquer avec l'utilisateur

- UIAlertView
- Notifications locales
- Présentation des notifications push
  - - Atelier : Ajouter des notifications, changer l'état de l'application en fonction

## Le graphisme sous iOS

- Ressources image
- Animations
  - - Atelier : Réalisation d'un composant personnalisé

## Le réseau et le Web

- L'état du réseau
- Le client Http : requête GET, POST
- L'accès aux services Web
- Le flux JSON
  - - Atelier : transfert de fichiers avec un serveur, exploitation de flux JSON via un service Web

## Publication d'application

- Préparation de l'application : principes, gestion des versions, génération du paquetage et signature
- Comment fonctionnent les certificats

## Outils avancés de développement

- Les gestionnaires de dépendances : Carthage, CocoaPods, Swift Packet Manager
- Contrôler les performances de son application
  - - Atelier : Utilisation des différents outils