

PROPUESTA DE CAPACITACIÓN KMMX

CDMX a 13 de abril del 2022.

Curso Integral de Desarrollo de Aplicaciones iOS con Swift

Descripción:

Swift es un nuevo lenguaje de programación para Cocoa y Cocoa Touch. Escribir código es interactivo y muy intuitivo, la sintaxis es concisa y expresiva, y la velocidad de ejecución de aplicaciones es considerablemente mucho más rápida. Swift de lado a lado trabaja con Objective-C, con Swift, Apple pone a disposición de los desarrolladores muchas más funcionalidades y más libertad a la hora de realizar su trabajo gracias a más de 4.000 APIs, entre las que encontramos la integración con Touch ID, integración con Siri, la posibilidad de crear filtros para la aplicación nativa de la cámara o nuevas APIs para iCloud entre otras muchas. SwiftUI es el nuevo framework de Apple para la creación de vistas que viene a sustituir al antiguo 'UIKit'. Este framework está diseñado para aprovechar todo el potencial de 'Swift', ya que ha sido creado desde cero teniendo en cuenta todo lo que este lenguaje nos proporciona. Se trata de un framework exclusivo de 'Swift', por lo que no puede ser usado en 'Objective-c'. No obstante, hay mecanismos para poder presentar las vistas desde 'Objective-c' en caso de ser necesario.

Objetivo del curso:

Al finalizar el Curso de Introducción al Desarrollo de aplicaciones para iOS con Swift, el alumno tendrá los conocimientos y habilidades para desarrollar con Swift aplicaciones para iOS (iPod Touch, iPad, iPhone) altamente optimizadas, ágiles, veloces, interactivas y funcionales. Aprenderá a: Identificar cómo y cuándo usar clases, estructuras y enumeraciones. Usar un diseño orientado a protocolos con extensiones para escribir código fácil de administrar. Aprovechar los patrones de diseño con Swift para resolver problemas de diseño comunes. Aplicar copia en escritura para sus tipos de valores personalizados para mejorar el rendimiento. Agregar simultaneidad a sus aplicaciones usando Grand Central Dispatch y colas de operaciones. Implementar genéricos para escribir código flexible y reutilizable. Reconocer el modelo de seguridad de iOS y los

límites de sus protecciones integradas. Reconocer las innumerables formas en que los datos confidenciales pueden filtrarse a lugares que no deberían. Implementar el cifrado con Keychain, la API de protección de datos y CommonCrypto. Identificar defectos heredados de C que todavía causan problemas en las aplicaciones modernas de iOS. Resolver cuestiones de privacidad relacionadas con la recopilación de datos del usuario y cómo mitigar posibles dificultades

Temario:

Parte 1. Trabajando con Objective-C

Cómo funciona la programación con Objective-C | Objective-C y Foundation | Clases, Herencia y otros conceptos desde el punto de vista de Objective-C | Escribiendo archivos con NSString y NSData | Aplicaciones por eventos controlados

Parte 2. Introducción a Swift

Introducción | Lo que Swift necesita | Cambios respecto a Objective-C | Desventajas de Objective-C | Características generales de Swift

Parte 3. Swift Tools

Xcode | Características de Xcode para Swift | Instalando y ejecutando Xcode | Explorando Playgrounds | Creando un nuevo proyecto en Xcode

Parte 4. Fundamentos de Swift

Estructura básica de un programa en Swift | Swift Playground | Tu primer Hola Mundo en Swift | Adición y substracción en Swift | Multiplicación y división en Swift | Construyendo y compliando aplicación

Parte 5. Variables y tipos de dato en Swift

Constantes y variables | Tipos básicos | Tipo Alias | Tuplas | Strings | Arrays | Diccionarios | Diferencias entre variables en Swift respecto a Objective-C

Parte 6. Operadores

Términos básicos | Operador de asignación | Operador aritmético | Operadores de incremento y decremento | Operadores de asignación compuestos | Operador de comparación | Operadores de rango | Operadores lógicos

Parte 7. Controlando el flujo del programa

Control de flujo | Declaración de ciclos | Declarando un For | Declarando un For In | While | Do-While | Declaración de ramificaciones | Declaración de un If | Switch | Declarando transferencia de control

Parte 8. Funciones

Funciones | Tipos de función | Funciones anidadas

Parte 9. Clases y Estructuras

Comparando Clases y Estructuras | Definición de sintaxis | Clases y Estructuras instanciadas | Accediendo a Propiedades | Inicializadores para tipos de Estructura | Tipo Value | Tipos de referencia | Ejemplo real con Estructuras y Clases

Parte 10. Propiedades, Métodos y Herencia

Propiedades | Propiedades alojadas | Propiedades creadas | Métodos | Instanciando Métodos | Definiendo una Clase base | Subclases | Remplazando y previniendo sobreescritura

Parte 11. Enumeración

Valores asociados | Raw Value | Implementando enumeración

Parte 12. Manejo de memoria

Manejo incorrecto de memoria | Recuento automático de referencias | Ciclos de referencia entre instancias de Clase | Seguimiento de asignación y fugas de memoria

Parte 13. Protocolos y extensiones de protocolo

Protocolos como tipos | Polimorfismo con protocolos | Tipificación con protocolos | Extensiones de protocolo | ¿Necesito usar protocolos? | Adopción de protocolos utilizando una implementación sintetizada | Biblioteca estándar de Swift

Parte 14. Diseño orientado a protocolos

Requisitos | Diseño orientado a objetos | Diseño orientado a protocolos | Estructuras versus clases

Parte 15. Genéricos

Introduciendo genéricos | Funciones genéricas | Tipos genéricos | Subíndices genéricos | Tipos asociados

Parte 16. Manejo de errores y disponibilidad

Manejo nativo de errores | Detectando errores | Cláusulas de captura de múltiples patrones | El atributo de disponibilidad