1. DATOS GENERALES DE LA ASIGNATURA

Nombre de la asignatura:	Programación Móvil
Clave de la asignatura:	DAM-2404
SATCA[1]:	2– 4 - 6
Carrera:	Ingeniería en Sistemas Computacionales

2. PRESENTACIÓN

Caracterización de la asignatura

La asignatura permite al estudiante adquirir competencias específicas para diseñar y producir soluciones a problemas reales mediante aplicaciones móviles, enfocadas a plataformas populares en el mercado, considerando la integración de distintas tecnologías y lenguajes propios que permiten aprovechar los recursos de hardware y software.

En los/las estudiantes se proporciona la capacidad de elaborar aplicaciones móviles que pueden estar presentes en dispositivos como tabletas y Smartphones (teléfonos inteligentes). Estos componentes se denominan mecanismos que permiten simplificar el uso de productos tecnológicos de manera interactiva e intuitiva, facilitando ciertas tareas en la vida diaria. Los objetos más recurrentes para los equipos son: educación, científico, diversión, pasatiempo, deportes, médicas, consultas, turismo, videojuegos, entre muchas otras.

La importancia de la asignatura se basa en el desarrollo de aplicaciones móviles basado en habilidades técnicas valiosas, incluyendo el dominio de lenguajes de programación específicos para dispositivos móviles y el uso de frameworks y herramientas asociadas.

La asignatura se relaciona con las asignaturas de Fundamentos de Programación, Programación Orientada a Objetos, Programación Lógica y Funcional y Base de Datos.

Intención didáctica

Se recomienda que el enfoque de este curso sea fundamentalmente práctico, tocando aspectos teóricos que faciliten la comprensión formal de los temas a tratar.

La asignatura se dividió en 5 temas, de tal, manera que el estudiante en el primer tema conozca los conceptos generales de las plataformas móviles, los entornos de desarrollo y componentes de una aplicación.

En el desarrollo de la primera competencia proporciona una comprensión básica de las plataformas móviles, la programación fundamental y el desarrollo de aplicaciones móviles.

[1] Sistema de Asignación y Transferencia de Créditos Académicos

En la segunda competencia el/la estudiante podrán crear aplicaciones con un diseño responsivo y adaptativo, permitiendo incorporar animaciones y transiciones avanzadas, accesibilidad y experiencia del usuario en desarrollo móvil, personalización de vistas y componentes

La tercera competencia permite que el estudiante conozca los diferentes modelos arquitectónicos aplicados al desarrollo de aplicaciones móviles.

Durante la cuarta competencia, se ven aspectos relacionados con la persistencia de datos y la gestión de dependencias en el desarrollo de aplicaciones móviles, específicamente en el contexto de la plataforma Android.

Finalmente, en la quinta competencia proporciona a los/las estudiantes una comprensión básica de qué son las API´s, los diferentes tipos que existen y por qué son importantes en el desarrollo de aplicaciones móviles. Además, explora cómo comenzar a trabajar con ellas y la forma de consumirlas de manera efectiva.

Durante el desarrollo de las actividades programadas con fechas de entrega preestablecidas, el/la estudiante aprende a valorar el trabajo en el aula y en casa, comprendiendo que construye su hacer futuro. Lo anterior, lo impulsa a actuar de una manera profesional, reconociendo la importancia del conocimiento y de los hábitos del trabajo individual y en equipo. El estudiante también desarrolla precisión, curiosidad, imaginación, puntualidad, entusiasmo, interés, tenacidad, flexibilidad y autonomía.

3. PARTICIPANTES EN EL DISEÑO Y SEGUIMIENTO CURRICULAR DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Tecnológico Nacional de México, Campus San Martín Texmelucan, abril/mayo de 2024	Docentes de la academia de Ciencias de la Computación del Tecnológico Nacional de México, Campus San Martín Texmelucan.	Generación de diseño, consolidación y/o seguimiento curricular, basado en reuniones de Academia de Ciencias de la Computación correspondientes a los meses de abril y mayo de 2024.

4. COMPETENCIA(S) A DESARROLLAR

Competencia(s) específica(s) de la asignatura
Desarrolla aplicaciones para dispositivos móviles que den soluciones a problemas en el contexto social actual.

5. COMPETENCIAS PREVIAS

- Implementa sistemas de infraestructura de redes para dar solución a problemas que impliquen una correcta administración de recursos computacionales.
- Desarrolla programas de cómputo basados en modelado de objetos para resolver problemas reales en diferentes contextos.
- Implementa Sistemas de base de datos basadas en SQL para el almacenamiento

6. TEMARIO

No.	Temas	Subtemas
1.	Introducción al Desarrollo Móvil	1.1. Introducción a las plataformas móviles 1.2. Fundamentos de la programación 1.3. Entorno de desarrollo 1.4. Componentes de una aplicación
2.	Vistas en Desarrollo Móvil	2.1. Principios del paradigma declarativo para el diseño de Interfaces de usuario 2.2. Introducción a las Interfaces de Usuario en Aplicaciones Móviles 2.2.1. Contenedores: Box, Column, Row 2.2.2. Componentes Composables 2.2.3. Previews 2.2.4. Modificadores 2.2.5. Estados 2.3. Diseño Responsivo y Adaptativo en Plataformas Móviles 2.4. Animaciones y Transiciones Avanzadas en Interfaces Móviles 2.5. Accesibilidad y Experiencia del Usuario en Desarrollo Móvil 2.6. Personalización de Vistas y Componentes en Aplicaciones Móviles
3.	Modelos arquitectónicos (MVVM, MVP)	3.1. Introducción a los Modelos Arquitectónicos en Desarrollo de Aplicaciones Móviles 3.2. Modelo MVC (Modelo-Vista-Controlador) en Aplicaciones Móviles 3.3. Modelo MVP (Modelo-Vista-Presentador) y su Aplicabilidad en el Desarrollo Móvil 3.4. Modelo MVVM (Modelo-Vista-Modelo de Vista) en Proyectos Móviles 3.5. Patrón Clean Architecture y su Relevancia en Desarrollo Móvil

		3.6. Patrón VIPER (View, Interactor, Presenter, Entity, Routing) y su Uso en Aplicaciones Móviles
4.	Acceso y almacenamiento de datos con inyección de dependencias	4.1. Introducción a la persistencia de datos. 4.2. Shared preferences y acceso a archivos 4.3. Base de datos locales 4.4. Fundamentos y beneficios de la inyección de dependencias. 4.5. Librerías de Inyección de Dependencias en Plataforma Android 4.5.1. Dagger 4.5.2. Hilt
5.	Consumo de Apis	5.1. Introducción al consumo de Apis 5.1.1. ¿Que es una Api? 5.1.2. Importancia de consumir Apis en aplicaciones móviles 5.1.3. Instalación de dependencias para Kotlin y Compose 5.1.4. Bibliotecas para manejar solicitudes de red ((OkHttp, Retrofit, etc.). 5.2. Tipos de Apis 5.2.1. E-commerce 5.2.2. Música y multimedia 5.2.3. Mapas y geolocalización 5.2.4. Datos abiertos 5.2.5. Redes Sociales 5.2.6. Clima 5.2.7. Machine Learning 5.2.8. Productividad 5.3. Manejo de Solicitudes 5.3.1 Clases para manejar las llamadas a la APIs 5.3.2 Definición de métodos para obtener y enviar los datos 5.3.3 Llamada a las APIs

7. ACTIVIDADES DE APRENDIZAJE DE LOS TEMAS

1. Introducción al Desarrollo Móvil	
Competencias	Actividades de aprendizaje
Específica(s): Conocer e identificar las plataformas móviles, programación, entorno de desarrollo y componentes de una aplicación. **Genéricas:** *Instrumentales:* • Capacidad de análisis y síntesis. • Conocimientos básicos de la carrera. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. *Interpersonales:* • Capacidad crítica y autocrítica. • Trabajo en equipo. • Habilidades interpersonales. *Sistémicas:* • Capacidad de aplicar conocimientos en la práctica. • Habilidades de investigación. • Capacidad de aprender. • Capacidad de generar nuevas ideas (creatividad). • Habilidad para trabajar en forma autónoma.	• Identificar las plataformas móviles • Investigar los conceptos de programación móvil • Identificar los componentes del Entorno de desarrollo • Investigar los componentes de una aplicación
2. Vistas en desarrollo móvil	
Competencias	Actividades de aprendizaje
Específica(s): Conocer las diferentes vistas disponibles y cómo utilizarlas de manera efectiva es fundamental para crear interfaces de usuario atractivas y funcionales.	• Investigar los principios del paradigma declarativo para el diseño de Interfaces de usuario. • Instalar las herramientas de desarrollo.

Genéricas: *Instrumentales:* • Capacidad de análisis y síntesis. • Conocimientos básicos de la carrera. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. *Interpersonales:* • Capacidad crítica y autocrítica. • Trabajo en equipo. • Habilidades interpersonales. *Sistémicas:* • Capacidad de aplicar conocimientos en la práctica. • Habilidades de investigación. • Capacidad de aprender. • Capacidad de generar nuevas ideas (creatividad). • Habilidad para trabajar en forma autónoma.	• Desarrollar aplicaciones móviles cuya interfaz incluya etiquetas, botones, campos de cajas de dialogo, spinners, radio button, checkbox, tabs, menús, multimedia y que permitan enviar y recibir mensajes o llamadas. • Aplicar los conocimientos en el desarrollo del Proyecto integrador.
3. Modelos arquitectónicos (MVVM, MVP)	
Competencias	Actividades de aprendizaje
Específica(s): Conocer y aplicar los modelos arquitectónicos para el desarrollo de aplicaciones móviles. **Genéricas:** *Instrumentales:* • Capacidad de análisis y síntesis. • Conocimientos básicos de la carrera. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. *Interpersonales:* • Capacidad crítica y autocrítica.	• Investigar en qué consisten los modelos arquitectónicos MVC, MVP y MVVM, así como su aplicabilidad de. • Implementar el modelo que más se adapte (MVC, MVP o MVVM) al proyecto integrador. • Investigar en qué consisten los patrones Clean Architecture y VIPER, y su aplicabilidad en aplicaciones móviles. • Implementar el patrón que más se adapte (Clean Architecture o VIPER) al proyecto integrador.

• Trabajo en equipo. • Habilidades interpersonales. *Sistémicas:* • Capacidad de aplicar conocimientos en la práctica. • Habilidades de investigación. • Capacidad de aprender. • Capacidad de generar nuevas ideas (creatividad). • Habilidad para trabajar en forma autónoma.	
4. Acceso y almacenamiento de datos con inyección de dependencias	
Competencias	Actividades de aprendizaje
Específica(s): Identificar como aplicar el acceso y almacenamiento de datos a través de librerías de inyección de dependencias. **Genéricas:** *Instrumentales:* • Capacidad de análisis y síntesis. • Conocimientos básicos de la carrera. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. *Interpersonales:* • Capacidad crítica y autocrítica. • Trabajo en equipo. • Habilidades interpersonales. *Sistémicas:* • Capacidad de aplicar conocimientos en la práctica. • Habilidades de investigación. • Capacidad de aprender. • Capacidad de generar nuevas ideas (creatividad). • Habilidad para trabajar en forma autónoma.	• Investigar en que consiste la persistencia de datos e inyección de dependencias. • Aplicar Shared preferences, acceso a archivos y Base de datos locales. • Aplicar Librerías de Inyección de Dependencias (Dagger, Hilt) en Plataforma Android en alguna aplicación móvil.

5. Consumo de Apis	
Competencias	Actividades de aprendizaje
Específica(s): Integrar Apis externas en aplicaciones móviles, aplicando buenas prácticas de desarrollo y maximizando su funcionalidad. **Genéricas:** *Instrumentales:* • Capacidad de análisis y síntesis. • Conocimientos básicos de la carrera. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. *Interpersonales:* • Capacidad crítica y autocrítica. • Trabajo en equipo. • Habilidades interpersonales. *Sistémicas:* • Capacidad de aplicar conocimientos en la práctica. • Habilidades de investigación. • Capacidad de aprender. • Capacidad de generar nuevas ideas (creatividad). • Habilidad para trabajar en forma autónoma.	• Investigar las API´S externas utilizadas en aplicaciones móviles, utilizando Kotlin y compose. • Realizar un mapa mental de las diferentes bibliotecas para manejar solicitudes de red en aplicaciones móviles, como OkHttp y Retrofit. • Realizar un cuadro comparativo sobre las características principales de cada biblioteca (OkHttp y Retrofit), sus ventajas y desventajas. • Instalar las dependencias necesarias para Kotlin y Compose. • Instalar las bibliotecas para manejar solicitudes de red (OkHttp, Retrofit, etc.). • Crear las clases para manejar las llamadas a las Apis y definición de métodos para realizar operaciones como: la obtención y envío de datos. • Implementar el ViewModel para manejar datos de la Api seleccionada. • Usar LiveData para notificar cambios en los datos obtenidos de la API. • Integración de datos obtenidos de la API en la interfaz de usuario.

8. PRÁCTICA(S)

- Instalar y configurar el entorno de desarrollo móvil.
- Desarrollar aplicaciones móviles cuya interfaz incluya etiquetas, botones, campos de cajas de dialogo, spinners, radio button, checkbox, tabs, menús, multimedia y que permitan enviar y recibir mensajes o llamadas.
- Implementar un modelo arquitectónico para el desarrollo de aplicaciones.

- Aplicar Librerías de Inyección de Dependencias en Plataforma Android.
- Instalar las bibliotecas para manejar solicitudes de red.
- Consumo de Apis, para el desarrollo de aplicaciones móviles.

9. PROYECTO DE ASIGNATURA

El objetivo del proyecto que planteé el docente con esta asignatura es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando estas fases:

- Fundamentación: marco referencial (teórico, conceptual, contextual, legal) en el que se fundamenta el proyecto según un diagnóstico realizado, que permite a los estudiantes comprender la realidad o situación objeto de estudio para definir un proceso de intervención o diseño de un modelo.
- Planeación: según el diagnóstico en esta fase, los estudiantes que asesoran al docente diseñaron el proyecto; implica planificar un proceso de intervención empresarial, social o comunitario, diseñar un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- Ejecución: consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y especificas a desarrollar.
- Evaluación: es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

10. EVALUACIÓN POR COMPETENCIAS

La evaluación debe ser diagnóstica, formativa y sumativa por lo que se debe considerar el desempeño en cada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Presentación de reportes de búsqueda de información en fuentes bibliográficas o digitales de reconocido valor.
- Participación en actividades para demostrar la comprensión de los conocimientos adquiridos a través de las investigaciones anteriores, tales como participación en foros, etc.
- Elaboración de proyectos de aplicación donde se incluyan e integren los algoritmos y modelos de minería de datos y aprendizaje automático.

- Reportes escritos de las prácticas realizadas, así como de las conclusiones obtenidas de dichas prácticas.
- Solución de casos de estudio o problemas.
- Desarrollo de proyectos integradores.

Instrumentos y herramientas:

- Rúbricas.
- Lista de cotejo.
- Guías de observación

11. FUENTES DE INFORMACIÓN

- CursoKotlin (s.f.). Curso programación Android en Kotlin. https://cursokotlin.com/curso-programacion-kotlin-android/
- Kotlin. (25 de marzo de 2019). Kotling Funciones.https://kotlin.desarrollador-android.com/empezando-con-kotlin/funciones/funciones/
- Juice Studio. (4 de enero de 2021). MVC vs MVVM vs MVP vs VIPER: ¿Qué arquitectura es adecuada? https://juice-studio.com/mvc-vs-mvvm-vs-mvp-vs-viper-que-arquitectura-de-diseno-es-adecuada/
- Bascón Pantoja, E. (2004). El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing. http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1683-07892004000100005
- Tecnología Móvil. (2024). ¿Cuáles son las prácticas recomendadas para administrar las dependencias en la aplicación móvil? [Post]. LinkedIn. https://www.linkedin.com/advice/0/what-best-practices-managing-dependencies-your-r36cc?lang=es&originalSubdomain=es