



PLANIFICACIÓN DE ACTIVIDAD CURRICULAR AÑO 2024

- **CARRERA:** Ingeniería En Informática
- **CÁTEDRA:** Arquitectura de Aplicaciones Móviles
- **RESPONSABLE DE CÁTEDRA:** Ing. Fabio D. Argañaraz A.

1. DATOS DE LA ACTIVIDAD CURRICULAR

Denominación:	Arquitectura de Aplicaciones Móviles	PLAN:	2022
Régimen:	Cuatrimestral	Carácter:	Obligatoria
Ubicación de la Asignatura en el Plan de Estudios:	5to Año - Cuatrimestral		
Correlativas anteriores:	Débiles: <ul style="list-style-type: none">• Programación I Fuerte: <ul style="list-style-type: none">• Base de Datos II• Programación II		
Correlativas posteriores:	Es correlativa fuerte de <ul style="list-style-type: none">• Trabajo Final		
Carga Horaria Semanal:	6hs		

2. COMPOSICIÓN DE EQUIPO DE CÁTEDRA

2. 1. Equipo de Cátedra

Apellido y Nombres	Grado académico máximo	Cargo docente	Dedicación en horas semanales al cargo
Fabio Damian, Argañaraz Azua	Ingeniero en Informática	Adjunto	6 hs.

2. 2. Auxiliares Alumnos

No posee

3. FUNDAMENTACIÓN

La materia **cuatrimestral** de “**Arquitectura de Aplicaciones Móviles**” en la carrera de **Ingeniería en Informática** de la Universidad Católica de Santiago del Estero (UCSE) tiene como objetivo proporcionar a los estudiantes los conocimientos teóricos y prácticos necesarios para diseñar y desarrollar aplicaciones móviles modernas, enfocándose en tecnologías multiplataforma utilizando **React Native**.

El desarrollo de aplicaciones móviles es una habilidad cada vez más demandada debido a la proliferación de dispositivos móviles y al aumento del consumo de aplicaciones en estos entornos. La elección de **React Native** como tecnología central responde a la necesidad de formar profesionales capacitados en



herramientas modernas y eficientes que permiten el desarrollo simultáneo para **Android** e **iOS**, aprovechando conocimientos previos en **React** y **JavaScript**, ya adquiridos en materias correlativas.

Durante el cursado, los estudiantes iniciarán con una **introducción al desarrollo móvil**, donde se presentarán las principales características de las aplicaciones móviles, su arquitectura y las diferencias entre aplicaciones nativas, híbridas y PWA (Progressive Web Apps). Se hará énfasis en la configuración de un entorno de desarrollo profesional, utilizando herramientas como **Node.js**, **React Native CLI** o **Expo**, permitiendo la rápida implementación y prueba de aplicaciones en emuladores y dispositivos reales.

En el núcleo del curso, se abordarán los **principios fundamentales de React Native**, haciendo especial hincapié en la creación de **interfaces de usuario** utilizando componentes como **View**, **Text**, **ScrollView** y **FlatList**, así como el manejo de estilos con **StyleSheet**. Los estudiantes aprenderán a trabajar con **eventos**, **navegación avanzada** (utilizando **React Navigation**) y gestión de estado, introduciendo conceptos clave como el manejo de **hooks** (**useState**, **useEffect**) y **contextos** globales (**useContext**).

A medida que el curso avance, se introducirá el **consumo de datos externos** mediante APIs REST, utilizando **fetch** o **Axios**, así como la implementación de manejo de errores y estados de carga. Además, se enseñarán estrategias para persistir información en el dispositivo móvil, utilizando herramientas como **AsyncStorage** y bases de datos locales como **SQLite**. Estas prácticas prepararán a los estudiantes para desarrollar aplicaciones con capacidad de funcionamiento **offline** y sincronización de datos.

Un aspecto central de la materia es el diseño y la **optimización de rendimiento**, abarcando prácticas recomendadas en React Native, la implementación de **animaciones fluidas** y principios de diseño **UX/UI** con herramientas como **Figma**. Se trabajará en la integración con **hardware del dispositivo**, cubriendo funcionalidades como GPS, cámara y almacenamiento de imágenes, y se enseñará la implementación de **notificaciones push** con **Firebase Cloud Messaging (FCM)** o **Expo Notifications**.

Finalmente, la materia abordará aspectos clave de la **seguridad** y la preparación para producción, cubriendo autenticación segura mediante **JWT** y la creación de builds para su despliegue en **Google Play Store** y **Apple App Store**. Se introducirá además el **testing** en aplicaciones React Native, utilizando herramientas como **Jest** para garantizar la calidad del código.

Como eje integrador del curso, los estudiantes desarrollarán un **proyecto práctico final** en grupos, que consistirá en la creación de una aplicación móvil completa con funcionalidades reales, navegación avanzada, consumo de APIs, almacenamiento local y optimización UX/UI. El proyecto se desarrollará en **entregas parciales** con feedback docente, permitiendo una construcción progresiva y colaborativa de la solución.

En resumen, **“Arquitectura de Aplicaciones Móviles”** es una materia cuatrimestral diseñada para brindar a los estudiantes una formación integral y práctica en el desarrollo de aplicaciones móviles modernas. A través de **React Native** y su ecosistema, los futuros profesionales adquirirán habilidades técnicas y metodológicas que les permitirán responder a los desafíos actuales del desarrollo multiplataforma, asegurando aplicaciones de alto rendimiento, seguras y con una experiencia de usuario óptima.

4. RELACIÓN DE LA ASIGNATURA CON EL PERFIL DE EGRESO

La materia **“Arquitectura de Aplicaciones Móviles”** está estrechamente vinculada con el **perfil de egreso del Ingeniero en Informática**, ya que aborda habilidades y conocimientos necesarios para el diseño, desarrollo y despliegue de sistemas móviles eficientes, modernos y multiplataforma. A continuación, se detallan las principales conexiones:

- **Habilidad para desarrollar aplicaciones móviles:** La materia proporciona los conocimientos teóricos y prácticos para que el estudiante pueda desarrollar aplicaciones móviles utilizando React Native. Esto se alinea con el perfil del Ingeniero en Informática, que debe ser capaz de diseñar, desarrollar y evaluar sistemas computacionales adaptados a las necesidades actuales del mercado tecnológico.
- **Desarrollo multiplataforma eficiente:** El uso de tecnologías como React Native permite a los estudiantes desarrollar aplicaciones compatibles tanto para Android como para iOS, optimizando recursos y tiempos. Esta habilidad es fundamental en un entorno profesional que demanda soluciones ágiles y eficientes, contribuyendo a la capacidad del egresado de responder a desafíos tecnológicos con herramientas modernas.
- **Conocimientos en experiencia del usuario (UX/UI):** A través del desarrollo de interfaces móviles y la incorporación de herramientas de prototipado como Figma, los estudiantes adquirirán habilidades para mejorar la usabilidad y accesibilidad de las aplicaciones. Esto fortalece el perfil del Ingeniero en Informática como profesional capaz de crear soluciones centradas en el usuario final, garantizando productos funcionales y de alta calidad.
- **Consumo e integración de APIs:** La materia capacita a los estudiantes en la integración de APIs REST y en el manejo de datos mediante herramientas como fetch y Axios, así como en la persistencia local con SQLite y AsyncStorage. Estas competencias son fundamentales para el desarrollo de sistemas complejos y conectados, alineados con las necesidades de la industria actual.
- **Uso de herramientas y tecnologías modernas:** La incorporación de herramientas como React Native CLI, Expo, Firebase, Jest y React Navigation asegura que el estudiante se forme en tecnologías actualizadas y ampliamente utilizadas en el desarrollo de aplicaciones móviles. Esto permite que el Ingeniero en Informática esté capacitado para integrarse rápidamente a equipos de trabajo y proyectos en la industria tecnológica.
- **Optimización y seguridad en aplicaciones:** A lo largo del curso, los estudiantes trabajarán en la optimización del rendimiento de aplicaciones, el manejo de errores, la implementación de autenticación segura con JWT y la creación de builds para producción. Estas habilidades contribuyen al desarrollo de sistemas robustos, escalables y seguros, alineándose con el perfil profesional del egresado.
- **Integración de hardware y funcionalidades avanzadas:** La implementación de funcionalidades nativas como GPS, cámara, almacenamiento de imágenes y notificaciones push fortalece las competencias del estudiante en la integración de hardware y software, una habilidad esencial para un Ingeniero en Informática en la era de los dispositivos móviles e IoT.

5. RELACIÓN DE LA ASIGNATURA CON LOS ALCANCES DEL TÍTULO.

La asignatura “**Arquitectura de Aplicaciones Móviles**”, perteneciente a la carrera de **Ingeniería en Informática**, es clave para el desarrollo de conocimientos y competencias avanzadas en el diseño, implementación y despliegue de aplicaciones móviles multiplataforma, cumpliendo con los **alcances del título** de Ingeniero en Informática. A través de esta asignatura, se abordan conceptos fundamentales y herramientas modernas que permiten a los futuros profesionales enfrentar los desafíos del desarrollo móvil actual.

La asignatura comienza con una **introducción al desarrollo móvil**, donde los estudiantes comprenderán la arquitectura de las aplicaciones móviles y las diferencias entre aplicaciones **nativas, híbridas y multiplataforma**. Esta base teórica es fundamental para diseñar soluciones alineadas con los requerimientos de la industria actual.

A nivel **técnico**, se enseña el desarrollo de aplicaciones utilizando **React Native**, un framework que permite la creación eficiente de aplicaciones para **Android e iOS** desde una misma base de código. Esta habilidad está directamente relacionada con la **capacidad del Ingeniero en Informática de diseñar, desarrollar e implementar sistemas de software eficientes y adaptables** a distintos entornos y dispositivos.

La materia también proporciona conocimientos en **consumo de APIs REST** y manejo de **almacenamiento local** mediante herramientas como **AsyncStorage** y **SQLite**, formando al estudiante en la integración de sistemas distribuidos y en el manejo de datos, que son pilares fundamentales del desarrollo de software robusto y escalable.

En cuanto a la **experiencia del usuario (UX/UI)**, se desarrollan competencias en el diseño de interfaces móviles intuitivas y accesibles, utilizando **Figma** y principios de diseño adaptativo. Estas habilidades son esenciales para garantizar que las soluciones desarrolladas sean **efectivas, usables y centradas en el usuario final**.

La asignatura aborda, además, **temas avanzados** como la integración de **hardware del dispositivo** (GPS, cámara, almacenamiento), el uso de **notificaciones push** con herramientas como **Firebase**, y el trabajo **offline** mediante sincronización de datos locales. Estas competencias son críticas en el desarrollo de aplicaciones modernas que deben funcionar en escenarios diversos y garantizar **rendimiento, seguridad y disponibilidad**.

Finalmente, se capacita a los estudiantes en **testing de aplicaciones** con **Jest** y en la preparación de builds para despliegue en **Play Store y App Store**, lo cual completa el ciclo de desarrollo y asegura la entrega de productos de **alta calidad** y alineados a los estándares del mercado.

6. COMPETENCIAS A LAS QUE SE APORTA

Competencias Genéricas

CG1. Capacidad de análisis y síntesis.

Analizarán y sintetizarán requerimientos funcionales y no funcionales para diseñar arquitecturas de aplicaciones móviles escalables y efectivas, considerando las diferencias entre tecnologías nativas y multiplataforma.

CG2. Capacidad de aplicar los conocimientos en la práctica.

Aplicarán los fundamentos de React Native en la creación de aplicaciones móviles completas, desde la configuración del entorno de desarrollo hasta el despliegue en tiendas oficiales.

CG3. Capacidad para aprender de forma autónoma.

Desarrollarán habilidades para explorar documentación técnica, guías oficiales de React Native y nuevas herramientas del ecosistema móvil, fomentando el aprendizaje continuo.

CG4. Habilidades para buscar, procesar y analizar información de diversas fuentes.

Buscarán soluciones en recursos como foros, documentación oficial y artículos técnicos para integrar APIs externas y resolver problemas específicos en el desarrollo móvil.

CG5. Capacidad de organizar y planificar.

Gestionarán proyectos de desarrollo móvil utilizando entregas parciales, cronogramas y herramientas de planificación ágil como Trello o Jira, asegurando el cumplimiento de objetivos.

CG6. Toma de decisiones responsable y ética.

Adoptarán prácticas éticas al trabajar con datos de usuarios, asegurando la implementación de medidas de seguridad como autenticación segura y manejo responsable de permisos en dispositivos móviles.

CG7. Habilidades interpersonales y trabajo en equipo.

Colaborarán en equipos multidisciplinarios para desarrollar el proyecto final, fomentando la comunicación efectiva y la resolución conjunta de problemas.

CG8. Comunicación oral y escrita efectiva.

Presentarán sus proyectos finales mediante exposiciones orales y reportes técnicos que expliquen las decisiones arquitectónicas y las funcionalidades implementadas.



Competencias Específicas

CE1. Diseño de soluciones de software adaptadas a dispositivos móviles.

Diseñarán arquitecturas de aplicaciones móviles considerando las limitaciones de hardware, consumo energético y experiencia del usuario, aplicando patrones de diseño adecuados.

CE2. Implementación de aplicaciones multiplataforma utilizando frameworks modernos como React Native.

Desarrollarán aplicaciones móviles funcionales para Android e iOS, optimizando el uso de componentes reutilizables y maximizando la eficiencia en el desarrollo.

CE3. Integración de hardware y software móvil para funcionalidades avanzadas.

Implementarán características como el uso de la cámara, GPS y almacenamiento de imágenes, utilizando APIs nativas y de terceros integradas con React Native.

CE4. Aplicación de principios de diseño UX/UI en interfaces móviles.

Diseñarán interfaces centradas en el usuario, utilizando herramientas como Figma y aplicando principios de diseño responsivo y accesible.

CE5. Optimización y pruebas de rendimiento en aplicaciones móviles.

Identificarán y resolverán cuellos de botella en el rendimiento mediante prácticas de optimización, incluyendo el uso de herramientas de profiling y testing automatizado con Jest.

CE6. Desarrollo seguro con manejo de autenticación y despliegue a producción.

Implementarán sistemas de autenticación segura con JWT y desarrollarán builds optimizadas para su despliegue en las tiendas de aplicaciones, siguiendo estándares de seguridad.

CE7. Uso de herramientas y tecnologías modernas para el desarrollo móvil.

Configurarán entornos de desarrollo profesional utilizando Node.js, React Native CLI, Expo y otras herramientas modernas, asegurando la calidad y eficiencia en el ciclo de desarrollo.

7. RESULTADOS DE APRENDIZAJE (RA)

RA1. Diseñar aplicaciones móviles eficientes y modernas para plataformas Android e iOS.

Aplicarán los principios de arquitectura móvil y React Native para desarrollar aplicaciones que sean funcionales, escalables y optimizadas para ambas plataformas, resolviendo problemas prácticos del mundo real.

RA2. Crear interfaces de usuario intuitivas y adaptativas utilizando principios de UX/UI.

Diseñarán y desarrollarán interfaces que respondan a las necesidades del usuario, utilizando herramientas como Figma y estilos personalizados en React Native, asegurando consistencia visual y accesibilidad.

RA3. Implementar consumo de APIs REST, manejo de errores y almacenamiento de datos.

Integrarán servicios externos mediante fetch o Axios, gestionarán errores de conexión o datos inválidos, y almacenarán información de manera persistente con AsyncStorage y SQLite para soportar funcionalidad offline.

RA4. Configurar entornos de desarrollo y realizar pruebas de aplicaciones móviles.

Prepararán entornos profesionales utilizando Node.js, Expo y React Native CLI, además de realizar pruebas unitarias y de integración con Jest para garantizar la calidad del software.

RA5. Optimizar aplicaciones para rendimiento y seguridad, incluyendo autenticación con JWT.

Implementarán buenas prácticas en el código para mejorar el rendimiento, protegerán las aplicaciones mediante autenticación segura con JWT y minimizarán vulnerabilidades comunes en el desarrollo móvil.

RA6. Integrar funcionalidades avanzadas como GPS, cámara, notificaciones push y sincronización offline.

Configurarán y utilizarán las APIs nativas de React Native para integrar estas funcionalidades avanzadas, mejorando la experiencia del usuario con aplicaciones dinámicas y adaptadas a escenarios reales.

RA7. Presentar y defender un proyecto final que cumpla con estándares industriales.

Desarrollarán, documentarán y defenderán un proyecto completo en equipo, justificando las decisiones técnicas y demostrando cómo el producto cumple con las mejores prácticas y requerimientos industriales.

MATRIZ DE VINCULACIÓN COMPETENCIAS/CONTENIDOS			
RA	COMPETENCIAS		UNIDADES DEL PROGRAMA ANALÍTICO
	GENÉRICAS	ESPECÍFICAS	
RA1	CG1, CG2, CG3, CG4	CE1, CE2	Unidad 1, Unidad 2
RA2	CG1, CG2, CG7	CE4	Unidad 2
RA3	CG2, CG3, CG4	CE3, CE7	Unidad 3
RA4	CG4, CG5, CG8	CE7	Unidad 1, Unidad 2, Unidad 3
RA5	CG1, CG2, CG5, CG6	CE5, CE6	Unidad 4, Unidad
RA6	CG2, CG4, CG5	CE3, CE4	Unidad 5
RA7	CG7, CG8	CE7	Unidad 6

RA1 (Diseñar aplicaciones móviles eficientes y modernas): Se relaciona con competencias genéricas como análisis y aplicación práctica de conocimientos (CG1, CG2), aprender de manera autónoma (CG3), y buscar información (CG4). Además, está vinculado con las competencias específicas de diseño de soluciones adaptadas a dispositivos móviles y la implementación de aplicaciones multiplataforma (CE1, CE2), abordando las primeras unidades sobre introducción y fundamentos de React Native.

RA2 (Crear interfaces intuitivas): Abarca las competencias genéricas relacionadas con análisis y aplicación de conocimientos (CG1, CG2), y trabajo en equipo (CG7), con énfasis en el diseño de interfaces UX/UI (CE4), que se estudian principalmente en la Unidad 2.

RA3 (Consumo de APIs y manejo de datos): Implica competencias para aplicar conocimientos en la práctica (CG2) y la búsqueda de información relevante (CG4). Aquí se aborda el consumo de APIs y almacenamiento de datos (CE3, CE7), contenido de la Unidad 3.

RA4 (Configurar entornos de desarrollo y pruebas): Relacionado con la capacidad de buscar, procesar y analizar información (CG4), planificar y organizar (CG5), y habilidades de comunicación (CG8). Se cubre con las herramientas y configuraciones en las Unidades 1, 2 y 3.

RA5 (Optimización de rendimiento y seguridad): Relacionado con la toma de decisiones responsables y éticas (CG6), planificación (CG5) y aplicar conocimientos en la práctica (CG2). Se aborda la optimización y pruebas de seguridad en la Unidad 4 y 6, vinculando las competencias específicas de pruebas de rendimiento y seguridad (CE5, CE6).

RA6 (Integración de funcionalidades avanzadas): Se vincula con la capacidad de aprender autónomamente (CG3), organizar y aplicar conocimientos (CG5), y utilizar tecnologías modernas (CE3, CE4). Se trabaja en la Unidad 5, donde se integran funcionalidades como GPS y cámara.

RA7 (Presentar y defender un proyecto final): Requiere habilidades interpersonales (CG7) y comunicación efectiva (CG8). Se aborda en la Unidad 6, donde los estudiantes desarrollan y presentan el proyecto final utilizando herramientas modernas (CE7).

8. CONTENIDOS MÍNIMOS (según Plan de Estudios vigente)

Desarrollo móvil. Configuración de entornos. Arquitecturas multiplataforma. Componentes y navegación en React Native. Gestión del estado. Consumo de APIs REST. Persistencia de datos. Integración con hardware móvil. Notificaciones push. Optimización de rendimiento. Diseño UX/UI. Seguridad en aplicaciones móviles. Testing. Despliegue

9. CRONOGRAMA DE DESARROLLO DE LAS UNIDADES TEMÁTICAS

UNIDADES TEMÁTICAS	Fechas estimadas para el desarrollo de las unidades	
	Desde	Hasta
1	Clase 1	Clase 3
2	Clase 4	Clase 9
3	Clase 10	Clase 14
4	Clase 15	Clase 18
5	Clase 19	Clase 22
6	Clase 23	Clase 26

10. DESARROLLO DE ACTIVIDADES PRÁCTICAS

Unidad	Tema/s	Actividades
1	Introducción al Desarrollo Móvil	Aprenderán sobre las características de las aplicaciones móviles, incluyendo las diferencias entre aplicaciones nativas, híbridas y PWA. Configurarán su entorno de desarrollo con herramientas como Node.js, Expo CLI y React Native CLI, creando su primer proyecto en React Native y ejecutándolo en emuladores y dispositivos reales.
2	Fundamentos de React Native	Se familiarizarán con los componentes básicos de React Native, cómo View, Text y Button, y aprenderán a trabajar con listas y scrolls usando ScrollView y FlatList. Además, implementarán eventos como onPress y onChangeText, gestionarán el estado con useState y useEffect, y desarrollarán aplicaciones con navegación avanzada utilizando React Navigation y gestión de estado con Context API.
3	Consumo de APIs y Manejo de Datos	Se abordará el consumo de datos mediante APIs REST utilizando fetch y Axios, y la persistencia de datos en el dispositivo con AsyncStorage y SQLite. Los estudiantes desarrollarán aplicaciones que consumen APIs externas, almacenan datos localmente y gestionan la sincronización de datos, permitiendo la funcionalidad offline.
4	Diseño y Optimización de Rendimiento	Aprenderán sobre diseño UX/UI y la importancia de la experiencia de usuario en aplicaciones móviles, utilizando herramientas como Figma para crear interfaces. Además, implementarán animaciones con React Native y optimizarán el rendimiento de sus aplicaciones mediante técnicas como VirtualizedList y lazy loading.
5	Integración de Hardware y Funciones Avanzadas	Integrarán funcionalidades avanzadas como GPS, cámara y almacenamiento de imágenes en sus aplicaciones, y aprenderán a implementar notificaciones push mediante Firebase Cloud Messaging. Realizarán prácticas donde combinarán estas funcionalidades para crear aplicaciones más completas y con mayor interacción con el dispositivo móvil.
6	Seguridad y Despliegue	Implementarán seguridad de las aplicaciones móviles mediante autenticación con JWT y la preparación de las aplicaciones para el despliegue en Google Play Store y Apple App Store. Los estudiantes también aprenderán a realizar testing con Jest para garantizar la calidad del código y presentarán sus proyectos finales, mostrando las aplicaciones completas que desarrollaron a lo largo del curso.



11. NÚMERO TOTAL DE HORAS CÁTEDRAS PROGRAMADAS (*conforme al Calendario Académico Vigente y en función de la carga horario Teórico-Práctica, establecida para su materia en el Plan de estudios*)

Período	Hs. Cátedra	Desarrollo		Total
		Teórico	Práctico	
1er. Cuatrimestre		43	44	87

12. METODOLOGÍA DE ENSEÑANZA

Unidad	Tema	Modalidad	Recursos didácticos
1	Introducción al Desarrollo Móvil	Clases teóricas y prácticas en laboratorio.	Presentaciones multimedia, tutoriales en video, guías de referencia, documentación oficial de React Native, plataformas para la configuración de entornos de desarrollo (Node.js, Expo, React Native CLI).
2	Fundamentos de React Native	Clases prácticas en laboratorio con seguimiento teórico.	Presentaciones multimedia, ejemplos prácticos, tutoriales interactivos, ejercicios en línea sobre React Native, documentación oficial, plataformas de pruebas (emuladores, dispositivos físicos).
3	Consumo de APIs y Manejo de Datos	Clases teóricas y prácticas en laboratorio.	Tutoriales de APIs REST, ejemplos de uso de Axios/fetch, guías de integración de APIs, plataformas de prueba (Postman, Expo), recursos sobre manejo de errores y almacenamiento de datos (AsyncStorage, SQLite).
4	Diseño y Optimización de Rendimiento	Clases teóricas y prácticas en laboratorio.	Herramientas de diseño (Figma, Sketch), tutoriales sobre diseño UX/UI, ejemplos de optimización de rendimiento en React Native, herramientas de pruebas de rendimiento (React DevTools, herramientas de navegación).
5	Integración de Hardware y Funciones Avanzadas	Clases prácticas con soporte teórico.	Tutoriales sobre integración con hardware (GPS, cámara, notificaciones push), guías sobre integración de Firebase Cloud Messaging (FCM), ejemplos prácticos de uso de APIs nativas de React Native.
6	Seguridad y Despliegue	Clases teóricas y prácticas en laboratorio.	Presentaciones sobre seguridad móvil y autenticación con JWT, guías de despliegue en Google Play Store y Apple App Store, recursos de testing y control de calidad (Jest, testing en React Native).



13. Programa Analítico

Unidad 1: Introducción al Desarrollo Móvil

- Introducción a las aplicaciones móviles
- Configuración del entorno de desarrollo
- Estructura básica de un proyecto en React Native

Unidad 2: Fundamentos de React Native

- Componentes básicos
- Listas y Scrolls
- Manejo de eventos
- Uso de hooks (useState, useEffect)
- Navegación avanzada
- Gestión de estado con Context API

Unidad 3: Consumo de APIs y Manejo de Datos

- Consumo de APIs REST con fetch
- Consumo de APIs con Axios
- Persistencia con AsyncStorage
- Uso de SQLite
- Sincronización de datos

Unidad 4: Diseño y Optimización de Rendimiento

- Diseño UX/UI y uso de Figma
- Animaciones con React Native
- Optimización de rendimiento
- Práctica integradora de diseño y animaciones

Unidad 5: Integración de Hardware y Funcionalidades Avanzadas

- Integración de GPS y geolocalización
- Uso de la cámara y almacenamiento de imágenes
- Notificaciones push con Firebase
- Práctica integradora de hardware y notificaciones

Unidad 6: Seguridad y Despliegue

- Autenticación segura con JWT
- Preparación de builds para Android e iOS
- Testing con Jest
- Presentación de proyectos finales

14. EVALUACIÓN Y ACREDITACIÓN

1.1. Requisitos para **PROMOCIONAR** la Asignatura

- El alumno debe tener las correlativas obligatorias aprobadas al momento del examen final de la asignatura.
- Poseer un mínimo 80% de asistencia de asistencia a clases teóricas y prácticas
- Los trabajos prácticos entregados en un Mínimo de 80% y aprobados.
- Todos los Parciales aprobados con 6 (seis) o más, con posibilidades de una sola instancia de recuperatorio por parcial.

1.2. Requisitos para **REGULARIZAR** la Asignatura

- El alumno debe tener las correlativas obligatorias aprobadas al momento del examen final de la asignatura.
- Poseer un mínimo 70% de asistencia de asistencia a clases teóricas y prácticas.
- Los trabajos prácticos entregados en un Mínimo de 70% y aprobados.
- Todos los Parciales aprobados con 5 (cinco) o más, con posibilidades de una sola instancia de recuperatorio por parcial.

1.3. Cronograma de Evaluaciones

Evaluación	Fecha Prevista
Evaluación Teórico Práctica	<i>Mitad de curso</i>
Proyecto Integrador / Exposición	<i>Final de Curso</i>

1.4. Rubricas de evaluación de Resultados de Aprendizajes (en caso de que corresponda).

Criterios de evaluación:

- o Comunicación y colaboración en equipo.
- o Calidad de código: legibilidad, mantenibilidad, organización.
- o Funcionalidad: el proyecto cumple con los requisitos especificados.
- o Diseño: la interfaz de usuario es intuitiva y atractiva.
- o Uso de herramientas de desarrollo: control de versiones, gestor de paquetes, etc.
- o Cumplimiento de plazos: el equipo entregó el proyecto en tiempo y forma.

Escala de valoración:



- Excelente: el equipo demostró excelencia en todos los criterios.
- Bueno: el equipo demostró un buen nivel en todos los criterios.
- Suficiente: el equipo demostró un nivel suficiente en todos los criterios, pero puede mejorar en algunos aspectos.
- Insuficiente: el equipo no alcanzó los criterios mínimos en uno o más aspectos.

1.5. Requisitos para Rendir Examen Final LIBRE

El estudiante para poder acceder al examen libre deberá presentar un trabajo de campo integrador con las consignas provistas por los docentes. Luego debe rendir y aprobar un examen escrito práctico integrando los conocimientos y habilidades de las prácticas realizadas durante el cursado de la asignatura. Luego, solo en el caso de haber aprobado la parte práctica, debe superar la instancia teórica que contempla todas las unidades temáticas.



15. BIBLIOGRAFÍA

1.6. Bibliografía Básica Obligatoria

Autor (es)	Título	Año de edición	Editorial	Ejemplares disponibles en bibl.

1.7. Bibliografía de Consulta

Autor (es)	Título	Año de edición	Editorial	Ejemplares disponibles en bibl.



16. PROPUESTA DE ACTIVIDADES DE EXTENSIÓN DE CÁTEDRA

Hackathon: se puede organizar una competencia de programación en la que los estudiantes tendrán que resolver problemas en un tiempo determinado. Esta actividad les permitirá poner en práctica sus habilidades de programación y trabajo en equipo.

Charlas y talleres: se pueden organizar charlas y talleres en los que los estudiantes presenten sus proyectos o aprendizajes en la materia de programación. Esto les permitirá desarrollar sus habilidades de comunicación y exposición, además de compartir sus conocimientos con otros compañeros.

Proyectos sociales: se pueden promover proyectos de programación que tengan un impacto social positivo en la comunidad. Por ejemplo, se puede desarrollar una aplicación que ayude a personas con discapacidades o una plataforma de donaciones para organizaciones sin fines de lucro. Esta actividad les permitirá a los estudiantes aplicar sus conocimientos en un contexto real y generar un impacto positivo en la sociedad.

Visitas a empresas de tecnología: se pueden organizar visitas a empresas de tecnología en las que los estudiantes puedan conocer de primera mano cómo se aplican los conocimientos de programación en el mundo laboral. Esto les permitirá tener una visión más clara de las oportunidades laborales que existen en el mercado y cómo se aplican los conocimientos adquiridos en la materia en el mundo real.

Participación en eventos de desarrollo web: se puede motivar a los estudiantes a participar en eventos de desarrollo web, como CACIC, JAIIIO, ekoparty, etc. , en los que puedan participar de talleres o como asistentes, así como también conocer a otros estudiantes y profesionales del área.