| **PLANIFICACIÓN DE CÁTEDRA** |
| --- |
| **2 0 2 5**   | ***logoTUP*** | **TECNICATURA**  **UNIVERSITARIA EN**  **PROGRAMACIÓN**  ***UTN – F.R. Resistencia***  *[SEDE: RESISTENCIA]* | | --- | --- | |
| **Asignatura:** **PROGRAMACIÓN III**  **Nivel:** 2025 PRIMER CUATRIMESTRE  **Carga horaria:** 8HS - SEMANALES.  **Coordinador de la Carrera**  Ing. Claudia Laclau  [tup@frre.utn.edu.ar](mailto:tup@frre.utn.edu.ar)  [www.frre.utn.edu.ar/tup](http://www.frre.utn.edu.ar/tup)  **Docente/s:**  *Comisión 1*  Plácido Emilio Maidana (Adjunto)  [placidomaidana@gmail.com](mailto:placidomaidana@gmail.com)  Emilio Puljiz (Categoría ayudante)  [emilio@vidadigital.com.ar](mailto:emilio@vidadigital.com.ar)  *Comisión 2*  Jose A. Fernandez (Adjunto)  [fernandezja@gmail.com](mailto:fernandezja@gmail.com)  Emilio Puljiz (Categoría ayudante)  [emilio@vidadigital.com.ar](mailto:emilio@vidadigital.com.ar)  *Reglamento de Estudio*: Ord. 1622/2018  *Diseño Curricular*: Ord. 2018/23   | 2CO | [www.frre.utn.edu.ar](http://www.frre.utn.edu.ar)  Tel 0362-4432928  French 414 (3500) Resistencia – Chaco | | --- | --- |   **-** |

**PLANIFICACIÓN DE CÁTEDRA**

**Fundamentación**

La materia Programación III está diseñada para brindar a los estudiantes una formación integral en el desarrollo de aplicaciones web y móviles, explorando el paradigma cliente-servidor y las tecnologías más actuales del mercado. A lo largo del curso los alumnos abordarán los temas mediante una experiencia práctica y completa, abarcando desde el diseño de la interfaz de usuario (Front-End) hasta la lógica del servidor (Back-End), la creación de APIs RESTful y el desarrollo de aplicaciones móviles nativas y responsiva, aplicando buenas prácticas de programación y estrategias de optimización para garantizar aplicaciones robustas, escalables y eficientes.

A lo largo de la materia, se utilizarán lenguajes de programación modernos y orientados a objetos, así como frameworks y herramientas actualizadas, para construir aplicaciones robustas, escalables y eficientes. Se hará especial énfasis en la aplicación de buenas prácticas de programación, el diseño de arquitecturas limpias y la optimización del rendimiento. A su vez, se fomentará el análisis y comprensión de los protocolos y servicios que intervienen en el proceso de desarrollo web, asegurando que los alumnos adquieran una visión integral del funcionamiento de las aplicaciones.

La materia abordará la gestión de bases de datos, explorando técnicas de mapeo objeto-relacional (ORM) y otras estrategias para simplificar la interacción con los datos. Además, se analizarán los protocolos y servicios involucrados en el desarrollo web, lo que permitirá a los estudiantes comprender a fondo el funcionamiento de las aplicaciones que construirán. Uno de los pilares fundamentales del curso será el desarrollo de APIs RESTful, permitiendo a los estudiantes crear servicios web que faciliten la integración entre diferentes aplicaciones, entornos, dispositivos y plataformas. Se introducirá a los alumnos en el desarrollo de aplicaciones móviles nativas y responsivas, promoviendo un enfoque "Mobile First" y priorizando la experiencia del usuario. Se fomentará el uso de metodologías ágiles y herramientas de control de versiones para gestionar el ciclo de vida del desarrollo de software de manera eficiente.

Por lo tanto, la materia Programación III no solo busca formar desarrolladores con habilidades técnicas avanzadas, sino también fomentar el pensamiento crítico, la capacidad de argumentación y la aplicación de buenas prácticas que les permitan abordar proyectos tecnológicos de manera eficiente y profesional.

**Objetivos Generales**

* Resolver problemas de aplicación en los que se evidencie la utilización de programación basada en arquitectura cliente-servidor.
* Desarrollar competencias en el diseño e implementación de aplicaciones web modernas, Incorporar tecnologías de desarrollo web tanto en el lado del servidor (Back-End) como del cliente (Front-End) mediante herramientas y frameworks actualizados.
* Argumentar en lenguaje coloquial y simbólico para explicar y justificar
* razonamientos y fundamentar procedimientos empleados en la resolución de los
* problemas relacionados
* Justificar decisiones tomadas en base a lineamientos técnicos
* Incorporar lenguaje técnico en presentaciones
* Fundamentar con evidencia y buena práctica
* Analizar el funcionamiento de protocolos y servicios involucrados en el proceso del desarrollo web.
* Fomentar habilidades para la creación de APIs RESTful escalables y eficientes: Aplicar las mejores prácticas y principios en el desarrollo de servicios web que faciliten la integración entre aplicaciones y plataformas.
* Introducir al desarrollo de aplicaciones móviles nativas y responsivas:Capacitar en el diseño y construcción de aplicaciones móviles utilizando tecnologías modernas, priorizando el enfoque "Mobile First" y la experiencia del usuario
* Promover el uso de metodologías y herramientas para la gestión de bases de datos: Utilizar mapeo objeto-relacional (ORM) y otras técnicas para simplificar la interacción con bases de datos, optimizando la productividad y mantenibilidad del código.

**Objetivos Específicos**

1. **Diseñar y desarrollar aplicaciones web escalables:**
   * Crear aplicaciones desde cero utilizando frameworks modernos de backend y frontend
   * Implementar servicios web que soporten arquitecturas escalables y eficientes.
   * Desarrollar aplicaciones web con comunicación a servicios API autenticados
2. **Desarrollar aplicaciones móviles nativas:**
   * Construir aplicaciones nativas y adaptativas utilizando frameworks modernos de desarrollo.
   * Ajustar los diseños al paradigma "Mobile First" para garantizar una experiencia de usuario óptima tanto en ambientes web como nativos mobile.
3. **Optimizar el uso de tecnologías de Front-End y Back-End:**
   * Integrar de manera eficiente herramientas de Front-End y Back-End en proyectos.
   * Implementar ORM para una interacción simplificada con bases de datos.
   * Diseñar funcionalidades completas, desde el modelo de datos hasta la interfaz de usuario con multiples servicios (web, api, escritorio, mobile)

**PROGRAMA ANALÍTICO**

**Unidad 1: Introducción a Aplicaciones web y móviles**

Contenidos:   
Aplicaciones Web:

Back end y front end, tecnologías del lado del servidor y tecnologías del lado del cliente. - Protocolo HTTP, métodos GET y POST, formularios y envío de parámetros al servidor.

Aplicaciones Móviles:

Fundamentos de Cliente-Servidor y Comunicación en Redes, Introducción al modelo cliente-servidor, Concepto de cliente y servidor, roles en una red. Protocolos comunes: HTTP, HTTPS, WebSocket. Difusión, unicast, multicast y broadcast. Tipos de comunicación en redes: Comunicación síncrona y asíncrona. APIs REST y GraphQL: diferencias y casos de uso. Introducción a WebSocket para comunicación en tiempo real. Procesos de solicitud y respuesta: Peticiones HTTP: métodos (GET, POST, PUT, DELETE, etc.). Cabeceras, cuerpo y códigos de estado. Cómo las aplicaciones móviles interactúan con APIs. Conceptos de backend básicos: Qué es un servidor, cómo procesa solicitudes. Bases de datos y su interacción con aplicaciones móviles. Introducción al concepto de servicios y endpoints. Seguridad básica: Importancia de HTTPS y encriptación. Tokens y autenticación: JWT (JSON Web Tokens). Buenas prácticas para proteger datos de usuario.

Ambiente Web. Separación en Capas. Nociones de Arquitecturas de Software . Uso de controladores y rutas. Concepto e implementación de Middlewares. Serialización de JSON. Control CORS. Gestión de Datos y Estado.

**Unidad 2: Desarrollo Web**

Contenidos:  
Lenguaje BackEnd/FrontEnd:

Fundamentos de programación en backend, frontend. Tecnologías para aplicar, entornos de desarrollo y despliegues. Aplicación práctica de cada uno. Programación Orientada a Objetos (clases, objetos, comportamiento, principios de herencia y polimorfismo. Interfaces. Inyección de dependencias)

Base de Datos (Relacional):

Creación y modificación de esquemas de base de datos (tablas, relaciones, índices). Uso de lenguajes de definición (DDL) y manipulacion (DML) . Consultas básicas (INSERT, UPDATE, DELETE, SELECT), consultas avanzadas (JOINS). Uso de cláusulas LIKE y BETWEEN. Funciones agregadas básicas: COUNT, SUM, AVERAGE.

Interfaces Responsivas:

Comprender y utilizar la grilla y la maquetación de vistas con frameworks actuales de estilos.

Introducción a JavaScript puro o “vanilla”

Fundamentos de JavaScript: Variables (let, const, var) y tipos de datos. Estructuras de control: condicionales, bucles (for, while, forEach). Funciones: declaración, expresiones y funciones flecha.

Manipulación del DOM: Selección y modificación de elementos HTML con JavaScript. Manejo de eventos: clic, teclado, carga de página.

Manejo de datos: Arrays y objetos: creación, manipulación y métodos importantes (map, filter, reduce). JSON: concepto, estructura y manipulación.

Conceptos avanzados: Closures y scope en JavaScript. Hoisting, event loop y modelo de ejecución. Promesas, async/await y manejo de errores (try...catch).

Módulos en JavaScript: Importación y exportación de módulos (import y export). Cómo organizar el código para un proyecto.

**Unidad 3: Desarrollo Móvil**

Contenidos:   
Framework backend especifico

Introducción a framework, rutas y controladores. - Creación de migraciones, vistas. - Uso de RM para CRUD y consultas avanzadas. - API RESTful: rutas, métodos HTTP, implementación y pruebas con Postman. - Middleware ,Funciones Principales de los Middleware ,Autenticación ,Autorización ,Protección Contra CSRF (Cross-Site Request Forgery) ,Registro de Actividad ,Control de Acceso a APIs ,Gestión de Sesiones. Buenas Prácticas de Seguridad. Gestión de sesiones en ambiente web. Validación y sanitización de datos. Prevención de ataques (OWASP TOP 10, por ejemplo inyecciones SQL y XSS)

Introducción a Desarrollo móvil

Qué es el Desarrollo nativo. Qué es un Framework para Desarrollo Móvil y por qué usarlo: Historia y diferencias con el Desarrollo Nativo. Diferencias con el desarrollo web. Estructura básica de un proyecto.

Configuración del entorno de desarrollo: Instalación de herramientas necesarias.Configuración de un emulador o pruebas en un dispositivo físico.

Componentes/librerías: Conceptos básicos. Ciclo de vida de los componentes específicos de la tecnología.

Primeros pasos con estilos: Aplicación de estilos para definir la apariencia de los elementos de la interfaz.

Desarrollo Móvil y Configuración del Entorno.

Configuración del entorno de desarrollo para aplicaciones móviles. Conceptos básicos de componentes móviles y su estructura. Uso de emuladores o simuladores para pruebas en diferentes plataformas. Gestión del estado y manejo de datos en la aplicación. Manejo de interacciones del usuario y visualización de listas.

Navegación y Manejo de Datos

Navegación en aplicaciones móviles: Instalación y configuración de React Navigation. Tipos de navegadores: stack, tab y drawer. Envío y recepción de parámetros entre pantallas.

Consumo de APIs y manejo de datos: Realización de peticiones HTTP para obtener o enviar datos. Procesamiento y visualización de datos en listas. Manejo de listas dinámicas.

Manejo de estado en la aplicación: Estado local con hooks (useState, useReducer). Introducción a Context API para compartir datos globales

Gestión Avanzada y Funcionalidades Extras

Estado global avanzado: Comparación entre Context API y Redux. Configuración y uso de Redux en React Native. Persistencia de datos en estado global.

Integración con APIs nativas: Uso de módulos nativos como cámara, ubicación y almacenamiento local. Introducción a las librerías react-native-camera y react-native-geolocation.

Animaciones y mejoras de UX: Implementación de animaciones con Animated API. Introducción a gestos con react-native-gesture-handler.

**Unidad 4: Publicación y Proyecto Final**

Contenidos:   
Preparación para producción General: Optimización del código y revisión de dependencias .Configuración de entornos de producción. Optimización de recursos. Implementación en servidores de nube (libres o con licencias para estudiantes).

Preparación para producción app movil: Generación de APK/IPA y pruebas finales. Requisitos para publicar en Google Play y Apple App Store.

Presentación del proyecto final

Creación de una aplicación completa con funcionalidad integrada:

Navegación multi-pantalla.

Consumo de API para mostrar y gestionar datos.

Uso de estado global.

Una funcionalidad nativa (por ejemplo, cámara o ubicación).

Publicación en entorno nube.

**ESTRATEGIAS METODOLÓGICAS**

* ABP: Aprendizaje Basado en Problemas: En el Aprendizaje Basado en Problemas (ABP), el profesor suministra información organizada con objetivos predefinidos y desarrolla la solución de problemas de ejemplo. El objetivo principal del ABP es que los estudiantes aprendan a través de la resolución de problemas, en lugar de solo memorizar información.
* La plataforma virtual del campus será utilizada para proporcionar material de estudio adicional, como lecturas, videos y ejercicios, así como para realizar evaluaciones en línea.
* Se utilizará la plataforma ZOOM para clases de apoyo, si fuera necesario. A través de ZOOM, los estudiantes podrán conectarse con el profesor y sus compañeros de clase para discutir temas relevantes y trabajar en proyectos conjuntos.

**CRONOGRAMA**

**CRONOGRAMA COMISION 1:**

Plácido Emilio Maidana (Adjunto), Emilio Puljiz (Categoría ayudante)

| **CLASE COMISION 1** | **FECHA** | **CONTENIDO A SER DESARROLLADO** |
| --- | --- | --- |
| 1 | Mar 18/03/2025 | Fundamentos de UX/UI en Mobile First:  Principios de diseño Mobile First.  Diferencias entre diseño web y diseño mobile.  Patrones de navegación en aplicaciones móviles. |
| 2 | Jue 20/03/2025 | Fundamentos de la ingeniería de software. Introducción a Aplicaciones web y móviles: Back end y front end, tecnologías del lado del servidor y tecnologías del lado del cliente.Entorno de Desarrollo AMP, Estructura de datos PHP.Lenguaje BackEnd: Fundamentos de programación en backend. |
| 3 | Mar 25/03/2025 | Introducción a Figma (o herramienta alternativa):Creación de cuentas y entorno de trabajo. Frames y diseño responsivo. Componentes reutilizables y diseño adaptativo. Prototipado interactivo: Creación de prototipos navegables. Animaciones y transiciones en Figma. |
| 4 | Jue 27/03/2025 | Protocolo HTTP, métodos GET y POST, formularios y envío de parámetros al servidor.Conceptos de programación sin estado en aplicaciones web.Programación Orientada a Objetos en PHP. |
| 5 | Mar 01/04/2025 | Diseñar pantallas de una aplicación de ejemplo (Home, Login, Registro). Crear un prototipo interactivo navegable. |
| 6 | Jue 03/04/2025 | Visión general del framework Laravel y su importancia en el desarrollo web moderno, Instalación y configuración del entorno de desarrollo con Laravel. Creación de proyectos y estructura de directorios. |
| 7 | Mar 08/04/2025 | Introducción a React Native: ¿Qué es React Native y cómo funciona? Diferencias con ReactJS y aplicaciones nativas.  Beneficios y casos de uso. Instalación y Configuración del Entorno: Instalación de Node.js y Expo CLI. |
| 8 | Jue 10/04/2025 | Desarrollo de un Blog con Laravel, Instalación de Composer, Instalación de Laravel y creación de proyecto,Flujo de Peticiones en Laravel , Instalación de Git SCM,Comandos Básicos de Git,Creación de un Formulario de Cálculo en Laravel,Implementar los Métodos del Controlador |
|  | Mar 15/04/2025 | 14/04 al 16/04 Exámenes Finales 1o llamado (con suspensión de clases para todos los niveles) |
|  | Jue 17/04/2025 | Jueves Santo (no laborable) |
| 9 | Mar 22/04/2025 | Configuración de Android Studio y Xcode para emulación.  Uso de dispositivos físicos para pruebas. Estructura de un Proyecto React Native: Carpetas y archivos principales. |
| 10 | Jue 24/04/2025 | Uso de Artisan, la interfaz de línea de comandos de Laravel. Conceptos fundamentales de MVC en Laravel. Creación y administración de rutas. |
| 12 | Mar 29/04/2025 | JSX y componentes funcionales.  Primeros Componentes: View, Text, Image, ScrollView.  Estilos en React Native: StyleSheet. Práctica: Crear una aplicación básica con pantalla de bienvenida y navegación inicial. |
|  | Jue 01/05/2025 | Día del trabajador |
| 12 | Mar 06/05/2025 | Componentes Reutilizables y Props: Creación de componentes personalizados. Uso de Props para pasar datos. Estado y ciclo de vida con Hooks (useState, useEffect). |
| 13 | Jue 08/05/2025 | Creación y modificación de esquemas de base de datos. Laravel: Migraciones y Eloquent ORM. Ventajas de las Migraciones. Estructura de una Migración. Comandos para Migraciones. Introducción a Eloquent ORM. Creación de Modelos en Eloquent. Asociación con Tablas en Eloquent. Consultas Básicas con Eloquent. Seguridad y Buenas Prácticas. Desarrollo de sistema escuela. Proyecto en Laravel, creación de migraciones y CRUD de las principales tablas del proyecto |
| 14 | Mar 13/05/2025 | Estilos Avanzados: Flexbox en React Native.  Estilos condicionales. Animaciones básicas con Expo.  Navegación con Expo: Instalación y configuración de React Navigation en Expo.Stack Navigator, Tab Navigator y Drawer Navigator. Manejo de rutas y Deep Linking en Expo. Pasar parámetros entre pantallas. Práctica: Cada grupo elige su propia aplicación a desarrollar. |
| **Parcial** | Jue 15/05/2025 | **Primer Examen parcial** |
| 15 | Mar 20/05/2025 | Implementar navegación con Expo en la app seleccionada. Crear componentes reutilizables y aplicar estilos avanzados. Definir el flujo de navegación de la aplicación elegida. |
| 16 | Jue 22/05/2025 | Tipos de Datos y Asociaciones en Laravel. Comandos Artisan y Migraciones en Laravel. Opciones adicionales del comando migrate. Ejecutar migraciones específicas. Revertir Migraciones. Actualizar Migraciones. Eliminar y recrear tablas.Opciones adicionales del comando migrate:generate. Impacto de las migraciones en los datos. Comandos migrate:refresh y migrate:fresh |
| 17 | Mar 27/05/2025 | Gestión de Estados Globales:  Introducción a Context API.  Manejo de estados globales con Context y Reducer.  Almacenamiento local con AsyncStorage.  Listas Dinámicas y FlatList: |
| 18 | Jue 29/05/2025 | Introducción a Blade. Definición de un Layout Base. Extender un Layout desde una Vista Secundaria. Mostrar Datos en Vistas. Control Estructural y Lógica en Blade. Autenticación y Roles.¿Qué es un ataque XSS? .Cómo evitar ataques XSS. |
| 19 | Mar 03/06/2025 | Consumo de APIs REST: ¿Qué es una API REST y cómo funciona? Fetch y Axios para solicitudes HTTP.  Manejo de respuestas y errores. |
| 20 | Jue 05/06/2025 | Implementación de controladores y modelos. Interfaces Responsive: Comprender y utilizar la grilla y la maquetación de vistas con frameworks actuales de estilos. Integración de Eloquent ORM. Validación de datos y manejo de formularios. |
|  | Mar 10/06/2025 | Exámenes Finales 2º llamado (con suspensión de clases para todos los niveles) |
| 21 | Jue 12/06/2025 | Uso del protocolo HTTP para mantener el estado entre servidor y cliente.Estrategias para el manejo de sesiones y cookies en Laravel. Relaciones en Laravel.Seeders. Crear Seeders. Editar los Seeders.Registrar Seeders. Ejecutar los Seeders |
| 22 | Mar 17/06/2025 | FlatList, SectionList. Carga infinita (Infinite Scroll). Pull to Refresh.  Práctica:  Consumir una API pública y mostrar datos en una lista.  Implementar búsqueda y filtrado de datos.  Integración con Backend (Firebase o Node.js):  Introducción a Firebase Realtime Database y Firestore.  Alternativa: Conexión con un backend en Node.js (Express).  Autenticación de Usuarios:  Registro y Login con Email/Contraseña.  Autenticación con redes sociales (Google, Facebook). |
| 23 | Jue 19/06/2025 | API RESTful. Protocolos y Estándares de API.REST.SOAP¿Cómo funcionan las API RESTful?Beneficios de las API RESTful.Métodos de autenticación RESTful.API y Postman.Tipos de Pruebas de API¿Cómo Analizar las Respuestas? Cómo Crear una API en Laravel.Respuesta en Formato JSON |
| 24 | Mar 24/06/2025 | Manejo de Tokens y Sesiones.  Protección de Rutas y Seguridad:  Rutas protegidas con navegación condicional.  Manejo de sesiones y persistencia con AsyncStorage.  Práctica:  Crear un flujo de autenticación completo con registro y login.  Proteger pantallas privadas (ej. Dashboard o Perfil). |
| 25 | Jue 26/06/2025 | Middleware ,Autenticación ,Autorización ,Protección Contra CSRF, Registro de Actividad ,Control de Acceso a APIs ,Gestión de Sesiones. Buenas Prácticas de Seguridad. Gestión de sesiones en ambiente web. Validación y sanitización de datos. Prevención de ataques |
| 26 | Mar 01/07/2025 | Repaso |
| Parcial | Jue 03/07/2025 | **Examen parcial** |
| Entrega | Mar 08/07/2025 | **Entrega y Exposición de trabajos prácticos** |
| Recuperatorios | Jue 10/07/2025 | **Recuperatorios** |

**CRONOGRAMA COMISION 2:**

Jose A. Fernandez (Adjunto), Emilio Puljiz (Categoría ayudante)

| **CLASE COMISION 2** | **FECHA** | **CONTENIDO A SER DESARROLLADO** |
| --- | --- | --- |
| 1 | Mar 18/03/2025 | Fundamentos de UX/UI en Mobile First:  Principios de diseño Mobile First.  Diferencias entre diseño web y diseño mobile.  Patrones de navegación en aplicaciones móviles. |
| 2 | Jue 20/03/2025 | Fundamentos de la ingeniería de software. Introducción a Aplicaciones web y móviles: Back end y front end, tecnologías del lado del servidor y tecnologías del lado del cliente. Entorno de desarrollo, Estructura de datos. Lenguaje BackEnd: Fundamentos de programación en backend. Programación .NET y Lenguaje C# |
| 3 | Mar 25/03/2025 | Introducción a Figma (o herramienta alternativa):Creación de cuentas y entorno de trabajo. Frames y diseño responsivo. Componentes reutilizables y diseño adaptativo. Prototipado interactivo: Creación de prototipos navegables. Animaciones y transiciones en Figma. |
| 4 | Jue 27/03/2025 | Protocolo HTTP, métodos GET y POST, formularios y envío de parámetros al servidor.Conceptos de programación sin estado en aplicaciones web.Programación Orientada a Objetos en el lenguaje de backend seleccionado. |
| 5 | Mar 01/04/2025 | Diseñar pantallas de una aplicación de ejemplo (Home, Login, Registro). Crear un prototipo interactivo navegable. |
| 6 | Jue 03/04/2025 | Visión general del framework de backend y su importancia en el desarrollo web moderno, Instalación y configuración del entorno de desarrollo. Creación de proyectos y estructura de directorios. |
| 7 | Mar 08/04/2025 | Introducción a React Native: ¿Qué es React Native y cómo funciona? Diferencias con ReactJS y aplicaciones nativas.  Beneficios y casos de uso. Instalación y Configuración del Entorno: Instalación de Node.js y Expo CLI. |
| 8 | Jue 10/04/2025 | Desarrollo de una aplicación en .NET (consola, web con MVC) y creación de proyecto, Flujo de peticiones, Instalación de Git.,Comandos básicos de Git. Creación de un Formulario de cálculo en web MVC. ,Implementar los Métodos del Controlador |
| 9 | Mar 15/04/2025 | 14/04 al 16/04 Exámenes Finales 1o llamado (con suspensión de clases para todos los niveles) |
|  | Jue 17/04/2025 | Jueves Santo (no laborable) |
| 10 | Mar 22/04/2025 | Configuración de Android Studio y Xcode para emulación.  Uso de dispositivos físicos para pruebas. Estructura de un Proyecto React Native: Carpetas y archivos principales. |
| 11 | Jue 24/04/2025 | Conceptos fundamentales de MVC. Creación y administración de rutas. |
| 12 | Mar 29/04/2025 | JSX y componentes funcionales.  Primeros Componentes: View, Text, Image, ScrollView.  Estilos en React Native: StyleSheet. Práctica: Crear una aplicación básica con pantalla de bienvenida y navegación inicial. Componentes Reutilizables y Props: Creación de componentes personalizados. Uso de Props para pasar datos. Estado y ciclo de vida con Hooks (useState, useEffect). |
|  | Jue 01/05/2025 | Día del trabajador |
| 13 | Mar 06/05/2025 | Estilos Avanzados: Flexbox en React Native.  Estilos condicionales. Animaciones básicas con Expo.  Navegación con Expo: Instalación y configuración de React Navigation en Expo. |
| 14 | Jue 08/05/2025 | Creación y modificación de esquemas de base de datos. Diseño preliminar de una DB. Consultas Basicas en SQL. Introduccion de acceso a datos via codigo ADO.NET (Connection, Command, DataReader). Conexion via ORM (EntityFramework, Dapper, RepoDB) Seguridad y Buenas Prácticas. Desarrollo de sistema escuela. Proyecto en .NET, creación de migraciones y CRUD de las principales tablas del proyecto |
| 15 | Mar 13/05/2025 | Stack Navigator, Tab Navigator y Drawer Navigator. Manejo de rutas y Deep Linking en Expo. Pasar parámetros entre pantallas. Práctica: Cada grupo elige su propia aplicación a desarrollar. |
| **Parcial** | Jue 15/05/2025 | **Primer Examen Parcial** |
| 16 | Mar 20/05/2025 | Implementar navegación con Expo en la app seleccionada. Crear componentes reutilizables y aplicar estilos avanzados. Definir el flujo de navegación de la aplicación elegida. |
| 17 | Jue 22/05/2025 | Tipos de Datos y Asociaciones en .NET. Conexion via ORM (EntityFramework, Dapper, RepoDB). Migración y diferentes formas de realizar Code First y DataBase First en EntityFramework. |
| 18 | Mar 27/05/2025 | Gestión de Estados Globales:  Introducción a Context API.  Manejo de estados globales con Context y Reducer.  Almacenamiento local con AsyncStorage.  Listas Dinámicas y FlatList: |
| 19 | Jue 29/05/2025 | Introducción a Motor de Vistas en desarrollo web. Ejemplos Razor. Ejemplo de ASP.NET MVC con Razor, Razor Pages y Blazor Estructura de componentes en motor de vista (Layouts, componentes) en Blade. Autenticación y Roles.¿Qué es un ataque XSS? .Cómo evitar ataques XSS. |
| 20 | Mar 03/06/2025 | Consumo de APIs REST: ¿Qué es una API REST y cómo funciona? Fetch y Axios para solicitudes HTTP.  Manejo de respuestas y errores. |
| 21 | Jue 05/06/2025 | Implementación de controladores y modelos. Interfaces Responsive: Comprender y utilizar la grilla y la maquetación de vistas con frameworks actuales de estilos. Validación de datos y manejo de formularios. |
| 22 | Mar 10/06/2025 | Exámenes Finales 2º llamado (con suspensión de clases para todos los niveles) |
| 23 | Jue 12/06/2025 | Uso del protocolo HTTP para mantener el estado entre servidor y cliente.Estrategias para el manejo de sesiones y cookies en .NET. Autenticacion oAuth, OpenID. Autenticacion basada en token |
| 24 | Mar 17/06/2025 | FlatList, SectionList. Carga infinita (Infinite Scroll). Pull to Refresh.  Práctica:  Consumir una API pública y mostrar datos en una lista.  Implementar búsqueda y filtrado de datos.  Integración con Backend (Firebase o Node.js):  Introducción a Firebase Realtime Database y Firestore.  Alternativa: Conexión con un backend en Node.js (Express).  Autenticación de Usuarios:  Registro y Login con Email/Contraseña.  Autenticación con redes sociales (Google, Facebook). |
| 25 | Jue 19/06/2025 | API RESTful. Protocolos y Estándares de API.REST.SOAP¿Cómo funcionan las API RESTful?Beneficios de las API RESTful.Métodos de autenticación RESTful.API y Postman.Tipos de Pruebas de API¿Cómo Analizar las Respuestas? Cómo Crear una API en NET.Respuesta en Formato JSON |
| 26 | Mar 24/06/2025 | Manejo de Tokens y Sesiones.  Protección de Rutas y Seguridad:  Rutas protegidas con navegación condicional.  Manejo de sesiones y persistencia con AsyncStorage.  Práctica:  Crear un flujo de autenticación completo con registro y login.  Proteger pantallas privadas (ej. Dashboard o Perfil). |
| 27 | Jue 26/06/2025 | Middleware ,Autenticación ,Autorización ,Protección Contra CSRF, Registro de Actividad ,Control de Acceso a APIs ,Gestión de Sesiones. Buenas Prácticas de Seguridad. Gestión de sesiones en ambiente web. Validación y sanitización de datos. Prevención de ataques |
|  | Mar 01/07/2025 |  |
| Parcial | Jue 03/07/2025 | **Examen parcial** |
| Entrega | Mar 08/07/2025 | Entrega y Exposición de trabajos prácticos |
| Recuperatorios | Jue 10/07/2025 | Recuperatorios |

**ORGANIZACIÓN DE ESPACIOS DENTRO Y FUERA DEL ÁMBITO UNIVERSITARIO**

Las clases serán de modo presencial y con apoyo del campus virtual, ‘www.cvfrre.com.ar’ medio por el cual se podrá acceder al material de apoyo. Aquí los alumnos podrán realizar consultas y sacarse las dudas que pudieran surgir fuera del horario de clases.

**RECURSOS NECESARIOS:**

Para un óptimo desarrollo de las clases prácticas, se requerirá una computadora personal con conexión a internet estable. Además, será indispensable contar con el siguiente software instalado:

* TODAS LAS COMISIONES
  + Git Bash: Para el control de versiones y la interacción con Git.
  + Node Package Manager (npm): Gestor de paquetes para JavaScript, utilizado para la instalación de dependencias del Front-End.
  + Un editor de código: (Visual Studio Code es muy recomendado)
  + Un navegador web: (Chrome, Firefox, etc.)
* COMISION 1
  + XAMPP: Para la gestión del servidor local, incluyendo Apache, MySQL y PHP.
  + Composer: Gestor de dependencias para PHP, esencial para la instalación y gestión de paquetes de Laravel.
  + Laravel: Framework PHP para el desarrollo de aplicaciones web modernas.
  + MySQL: Sistema de gestión de bases de datos relacional.
* COMISION 2
  + Visual Studio 2022 version Community (con modulo Web y de Escritorio)
  + SQL Server 2022 version Developer

Este conjunto de herramientas permitirá a los estudiantes abordar de manera integral el desarrollo de aplicaciones web y móviles, desde la creación del Back-End, hasta el Front-End con tecnologías modernas, y el control de versiones con Git.

**EVALUACION: Instrumentos y modalidad**

El resultado de la evaluación del estudiante estará expresado en números enteros dentro de la escala del UNO (1) al DIEZ (10). Para la aprobación de la asignatura se requerirá como mínimo SEIS (6) puntos.

**APROBACIÓN DIRECTA**

* Cumplir con los prerrequisitos de inscripción a la materia según diseño curricular.
* Asistir a clase cumpliendo los mínimos establecidos.
* Cumplir con las actividades de formación práctica.
* Aprobar el 100% de las instancias de evaluación teóricas, prácticas y/o ambas integradas. La materia cuenta con dos exámenes parciales, y un trabajo práctico, consistente en una aplicación.
* En el caso en que no haya aprobado alguna de las instancias de evaluación parcial, tendrá acceso a un examen recuperatorio.
* La calificación se expresa en número entero y en caso de promedios con decimales se redondeará al valor más próximo. La nota promedio de las instancias de evaluación aprobadas así obtenida será la calificación definitiva de aprobación de la asignatura.

**APROBACIÓN NO DIRECTA (Regularidad + Examen final)**

El estudiante que no alcanzó los objetivos de aprobación directa, pero que ha demostrado niveles mínimos y básicos de aprendizaje no alcance los objetivos de aprobación directa, estará habilitado a rendir evaluación final. Para rendir el examen final, deberá inscribirse conforme a los turnos de exámenes que convoque cada Facultad Regional.

Las condiciones mínimas requeridas para estar en condiciones de un examen final serán:

1. Haber realizado el 80% de las prácticas sugeridas por la cátedra.

2. Tener 75 por ciento de asistencia a las clases presenciales.

3. Tener aprobado ambos exámenes parciales para lo cual tiene acceso a dos recuperatorios

**NO APROBACIÓN (Libre)**

El estudiante que no ha demostrado niveles mínimos y básicos de aprendizaje deberá re-cursar la asignatura. No está habilitado a rendir examen final.

**CORRELATIVIDADES**

* Tener aprobada Programación II y Bases de Datos I
* Aprobada Programación I.

**BIBLIOGRAFÍA**

* Gilmore, W. J. (2018). Easy Laravel 5 A Hands On Introduction Using a Real-World Project. Lean Publishing, Febuary, 1-289.
* Index @ reactnative.dev. (n.d.). https://reactnative.dev/
* Laravel docs 11x. (2024). Laravel. https://laravel.com/docs/11.x
* Masse, M. (2012). REST API Design. In O'Reilly.
* Sweat, J. E. (2005). PHP Architect's Guide to PHP Design Patterns - Page 2. In Design. <http://books.google.com/books?id=hV_LyQwG0zAC&printsec=frontcover>