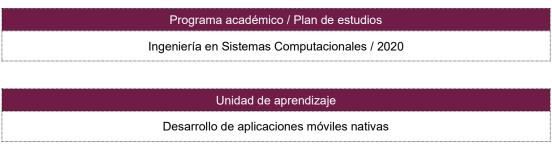


# Instituto Politécnico Nacional Escuela Superior de Cómputo





## Práctica 5: Consulta de Base de Datos vía APIs

**Objetivo**: El propósito de esta práctica es que los estudiantes desarrollen e integren consultas avanzadas a bases de datos mediante APIs, añadiendo funcionalidades de búsqueda y recomendaciones. La práctica debe incluir la persistencia de datos y la sincronización entre la base de datos local y las APIs remotas, asegurando que la aplicación mantenga la funcionalidad incluso sin conexión a internet.

# Ejercicio 1: Integración de Consultas a la Base de Datos vía API REST

**Descripción de la actividad:** Extienda la funcionalidad de su aplicación móvil para realizar consultas avanzadas a la base de datos utilizando endpoints de la API REST previamente creada y garantice la sincronización con bases de datos locales.

### Instrucciones específicas:

- 1. Conexión a la API REST:
  - o Conecte la aplicación móvil a la API REST desarrollada en prácticas anteriores.
  - Asegúrese de que los datos consultados puedan ser obtenidos y mostrados de forma eficiente en la aplicación.

# 2. Persistencia de la sesión:

 Mantenga la sesión del usuario activa y visible en la interfaz de usuario, similar a lo que se hace en redes sociales.

#### 3. Pruebas de conexión:

 Utilice Retrofit u otro cliente HTTP para confirmar la correcta conexión y funcionalidad de la API.

## Persistencia de Datos:

- Almacenamiento local: Implemente el uso de una base de datos local (SQLite, Room, etc.) para almacenar los datos obtenidos de las APIs y permitir la visualización sin conexión.
- **Sincronización de datos**: Desarrolle mecanismos que aseguren la sincronización de los datos entre la base de datos local y las APIs remotas para mantener la coherencia de la información.

# Ejercicio 2: Consumo de APIs Públicas

**Descripción de la actividad:** Integre el consumo de al menos una API de acceso gratuito en su aplicación, como la API de Open Library o TVMaze, para que los usuarios puedan realizar búsquedas y recibir recomendaciones.

#### Enlaces de interés:

- Open Library API: https://openlibrary.org/developers/api
- TVMaze API: <a href="https://www.tvmaze.com/api">https://www.tvmaze.com/api</a>

#### Requisitos:

# 1. Integración de APIs:

 Implemente al menos una de las APIs mencionadas para que la aplicación consuma y procese datos en formato JSON.

#### 2. Operaciones de Búsqueda:

- o Open Library: Implemente la funcionalidad para que los usuarios busquen libros y autores.
- o **TVMaze**: Implemente la funcionalidad para la búsqueda de series y películas.

## 3. Recomendaciones basadas en búsquedas y favoritos:

 Implemente recomendaciones personalizadas para el usuario en base a su historial y elementos guardados en favoritos.

# 4. Registro y Login con Google:

 La implementación de registro y login con Google es opcional. Se puede incluir la integración con otros servicios, como Facebook, de forma opcional.

# Persistencia de Datos y Sincronización:

- Almacenamiento de datos obtenidos: Garantice que los datos de las búsquedas y recomendaciones se almacenen en la base de datos local para permitir el acceso sin conexión.
- **Sincronización de datos**: Implemente mecanismos que aseguren la actualización y sincronización periódica de los datos entre la base de datos local y las APIs remotas.

## Ejercicio 3: Funcionalidades de Búsqueda, Favoritos y Recomendaciones

# Requisitos:

## 1. Historial y Agregación a Favoritos:

- Usuario: Implemente funcionalidades que permitan a los usuarios registrados ver su historial de búsquedas y agregar libros, autores, series o películas a su lista de favoritos.
- Administrador: Implemente la funcionalidad para que los administradores puedan visualizar los historiales y favoritos de todos los usuarios.

## 2. Sistema de Recomendaciones:

- Desarrolle un sistema de recomendaciones basado en los elementos guardados en favoritos y en el historial de búsquedas del usuario.
- Utilice la información almacenada para sugerir elementos relevantes y afines al usuario.

### Pruebas y Verificación:

- **Pruebas de API REST**: Realice pruebas de las funcionalidades implementadas, usando herramientas como Retrofit o Postman para confirmar la correcta integración y autenticación.
- **Persistencia de Datos**: Garantice que la información de las búsquedas, favoritos y recomendaciones se almacene de manera persistente en la base de datos.

# Entrega de la práctica:

## 1. Código fuente:

 Subir el código fuente de la aplicación móvil y de su Backend al repositorio de GitHub, garantizando que la conexión a la API y las funcionalidades de persistencia de datos estén implementadas y probadas.

### 2. Informe de la práctica:

- Portada: Incluya su nombre completo, número de boleta, nombre de la asignatura, nombre del profesor y fecha de entrega.
- Índice: Indique las secciones del informe.
- Desarrollo: Documente las funcionalidades implementadas, con capturas de pantalla y ejemplos de pruebas realizadas.

- Conclusiones: Indique los retos y logros principales alcanzados durante el desarrollo.
- o **Bibliografía**: Referencias consultadas en formato APA.

# Formato del documento:

• Tipo de letra: Arial.

Tamaño: 12.Espaciado: 1.5.

• Márgenes: Estándar.

**Fecha de Entrega:** La fecha límite para la entrega de esta práctica es el **martes 06 de Mayo de 2025**. No se aceptarán entregas fuera de tiempo y forma.