



# Instituto Politécnico Nacional

## Escuela Superior de Cómputo



Programa académico / Plan de estudios

Ingeniería en Sistemas Computacionales / 2020

Unidad de aprendizaje

Ingeniería de Software

### Práctica 3: Modelo de Análisis y Consumo de APIs en Proyectos de Software

**Objetivo:** El propósito de esta práctica es que desarrollen un modelo de análisis detallado de su software, incorporando diagramas y documentación que respalden la comprensión y estructura del proyecto mediante el uso de UML. Asimismo, deberán consumir servicios REST utilizando APIs públicas, procesando datos en formato JSON o XML e integrando funcionalidades adicionales a su software, asegurando la correcta conexión y funcionalidad con los endpoints implementados en prácticas anteriores.

#### Ejercicio 1: Desarrollo del Modelo de Análisis

**Descripción de la actividad:** Elabore el modelo de análisis para su software, detallando las funciones, interacciones y características clave que componen la solución. Utilice diagramas UML para representar y documentar su proyecto.

##### Instrucciones específicas:

##### 1. Documentación esencial mediante UML:

- **Diagrama Entidad-Relación (ER):** Si su software utiliza una base de datos, desarrolle un diagrama ER que represente las entidades y relaciones.
- **Diccionario de datos:** Describa los datos clave, detallando sus atributos y características.
- **Diagrama de casos de uso (UML):** Elabore un diagrama UML que represente los actores y las interacciones con el sistema.
- **Diagrama de secuencias del sistema (UML):** Describa el flujo del caso de uso más prioritario mediante un diagrama de secuencias.
- **Diagrama de clases conceptuales (UML):** Defina las clases y relaciones en el dominio de su software.
- **Modelo de interfaz y navegación:** Presente un esquema de la estructura de navegación de la interfaz de usuario.
- **Diagramas de robustez (UML):** Indique cómo se relacionan las clases, controladores y entidades en el sistema.
- **Diagrama de clases de diseño (UML):** Incluya detalles de atributos y métodos para cada clase.
- **Diagrama de interacción de clases (UML):** Explique la interacción entre las clases involucradas en los eventos principales.

##### Documentación FURPS+:

- **Funcionalidad (F):** Documente las capacidades y funcionalidades esenciales del software, como el manejo de usuarios, autenticación y consumo de APIs.
- **Usabilidad (U):** Incluya descripciones sobre la interfaz, la facilidad de uso y la experiencia de usuario.

- **Confiabilidad (R):** Especifique los mecanismos de autenticación, la encriptación de contraseñas y el manejo de errores.
  - **Rendimiento (P):** Defina los tiempos de respuesta aceptables y el uso de recursos del sistema.
  - **Soporte (S):** Documente aspectos relacionados con la mantenibilidad del sistema y la facilidad para realizar modificaciones.
  - **Extensiones (+):** Agregue consideraciones de diseño, implementación y restricciones físicas, como la compatibilidad con plataformas específicas y el uso de tecnologías particulares.
- 

## Ejercicio 2: Implementación de Funcionalidades Adicionales en su Software

### Requisitos:

1. **Foto de Perfil:**
    - **Usuario:** Permita que cada usuario suba y modifique su foto de perfil desde su cuenta.
    - **Administrador:** Habilite la capacidad de visualizar y modificar las fotos de perfil de todos los usuarios.
  2. **Conexión y Pruebas de la API REST:**
    - **Conexión con la aplicación:** Conecte su software con el servicio REST creado en la Práctica 2, manteniendo la funcionalidad de las operaciones CRUD y la gestión de roles.
    - **Pruebas de los endpoints:** Realice pruebas de cada endpoint usando Retrofit u otro cliente HTTP, verificando que el sistema de autenticación, roles y permisos funcione correctamente.
- 

## Ejercicio 3: Consumo de APIs Públicas

**Descripción de la actividad:** Integre al menos una API de acceso gratuito en su software, procesando y presentando la información al usuario. Se recomienda utilizar:

- **Open Library API:** Permite obtener datos sobre libros y autores. <https://openlibrary.org/developers/api>
- **TVMaze API:** Ofrece información sobre series y películas. <https://www.tvmaze.com/api>

### Requisitos:

- **Integración de APIs:** Debe consumir al menos una de las APIs mencionadas y procesar la respuesta JSON.
- **Operaciones:**
  - **Open Library:** Habilite la búsqueda de libros y autores.
  - **TVMaze:** Habilite la búsqueda de series y películas.

### Funcionalidades adicionales:

- **Consumo de más APIs:** Consuma ambas APIs u otras de su elección que cumplan con los requisitos.
  - **Registro y login con Google:** La implementación del registro y login mediante Google es obligatoria. La integración con otros servicios, como Facebook, es opcional.
- 

## Entrega de la práctica:

- Subir todo el código a su repositorio de GitHub.
- El informe debe incluir los siguientes apartados:
  - **Portada:** Nombre completo, número de boleta, nombre de la asignatura, nombre del profesor y fecha de entrega.
  - **Introducción:** Explicación general del código y la lógica utilizada.
  - **Desarrollo:** Código fuente y capturas de pantalla del sistema en funcionamiento.
  - **Conclusiones:** Retos y logros del desarrollo.
  - **Bibliografía:** Referencias en formato APA.
- **Formato:**
  - 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 miércoles 20 de noviembre de 2024. No se aceptarán entregas fuera de tiempo y forma.

Por favor, asegúrese de seguir estas instrucciones cuidadosamente para evitar cualquier inconveniente en la evaluación de su trabajo.

**NOTA:** Considere que el plagio es un delito. Cualquier evidencia de copia será sancionada con la anulación de ambas prácticas, la original y la copia