Universidad Autónoma de Zacatecas

**Unidad Ingeniería Eléctrica**

## Programa Ingeniería de Software

**Materia**

Análisis y Diseño Orientado a Objetos

**Docente**

Cristian Boyain

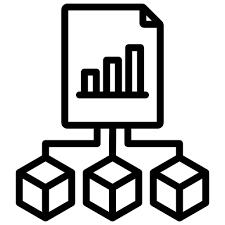
**Integrantes**

## José Manuel Roberto Badillo.

## Ángel Antonio de la Cruz Díaz.

## Diego Ignacio García Diaz de León.

## 4- Paul Rivera Salas.



### **Capítulo 6: Casos de Uso**

Este capítulo proporciona los pasos detallados para definir **Casos de Uso** en un formato comprensible y aplicable al contexto de un proyecto. Algunos puntos clave incluyen:

**Definición de Actores y Casos de Uso**:

Identificar actores (primarios, secundarios, o sistemas externos) y los objetivos que intentan lograr con el sistema.

Enfoque en casos de uso que representen **requisitos funcionales esenciales** del sistema.

**Motivación y Beneficios**:

Los casos de uso vinculan los requisitos con la funcionalidad esperada, ofreciendo un **contexto claro para el diseño posterior**.

**Formato Propuesto**:

Se recomienda usar un estilo "Fully Dressed" para detalles completos (actor, precondiciones, flujo principal, excepciones, postcondiciones).

**Directrices**:

**Escribir en estilo UI-Free**: No incluir elementos específicos de la interfaz de usuario para evitar limitar el diseño futuro.

**Escribir Casos de Uso Tersos y Black-Box**: Centrarse en la funcionalidad sin detallar la implementación interna.

### **Capítulo 7: Otros Requerimientos**

Este capítulo complementa los casos de uso abordando otros requisitos (no funcionales, reglas de negocio, etc.).

**Requisitos No Funcionales**:

Definir restricciones como rendimiento, seguridad, y capacidad de mantenimiento.

Integrar estos aspectos en el diseño técnico.

**Especificaciones Suplementarias**:

Detallar aspectos técnicos que no se reflejan directamente en los casos de uso (ejemplo: compatibilidad con hardware o estándares).

**Proceso Iterativo**:

Enfatiza que tanto los casos de uso como los requisitos deben evolucionar durante el desarrollo, alineándose con un enfoque iterativo y ágil.

**Relación con Reglas de Negocio**:

Las reglas de negocio identificadas deben integrarse en los flujos principales o alternativos de los casos de uso, proporcionando una **base para validaciones y excepciones**.

### **Aplicación al Proyecto**

**Definición de Casos de Uso (Capítulo 6)**:

Usa el formato detallado para representar los procesos como "Compra de Medicamentos" o "Gestión de Recetas Médicas".

Identifica claramente los actores y sus objetivos dentro de cada flujo.

**Requisitos Complementarios (Capítulo 7)**:

Asocia los requisitos no funcionales (ejemplo: tiempos de respuesta o almacenamiento seguro) con cada caso de uso relevante.

Incluye reglas de negocio (ejemplo: "Solo medicamentos con receta válida pueden venderse").

**Iteración y Diseño Evolutivo**:

Ajusta y refina los casos de uso y requisitos en cada iteración del diseño orientado a objetos, validando con actores clave.