



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

Universidad Nacional de Colombia - sede Bogotá  
Facultad de Ingeniería  
Departamento de Sistemas e Industrial  
Curso: Ingeniería de Software 1 (2016701)

## Restricciones del proyecto (Semana 15 → 14 a 18 de julio)

Desarrollar una aplicación funcional con una arquitectura monolítica que integre conceptos aprendidos durante el curso: Levantamiento y análisis de requerimientos, Clean Code, Patrones de Diseño, Testing, fundamentos de diseño y arquitectura de software. El proyecto debe contemplar todos los ciclos esenciales del desarrollo de software: análisis de requerimientos, diseño, desarrollo, pruebas y documentación básica.

## Reglas Generales del Proyecto

### 1. Arquitectura:

- Debe ser **monolítica**.
- No usar arquitecturas basadas en microservicios ni arquitecturas distribuidas.

### 2. Base de Datos:

- Debe utilizar una **base de datos relacional** (ej. PostgreSQL, MySQL, SQLite).

### 3. Tema del Proyecto:



- Lo ideal es generar diversidad en los temas de cada semestre, así que para este, las "chazas" u otras aplicaciones del mercado sin novedades que hagan un diferencial quedan descartadas.
- Se **descartan videojuegos**, Solo un grupo ha argumentado que es posible seguir la correcta implementación del ciclo de vida de software establecido a lo largo del curso, por lo que será un piloto.
- El proyecto debe resolver un problema **realista y acotado** con lógica de negocio clara.

### 4. Uso de APIs externas:

- Pueden **consumir APIs**, pero **no pueden exponerlas** (no hacer desarrollo de APIs REST públicas).
- Pueden **consumir APIs externas** únicamente como **apoyo complementario** al proyecto, pero **no deben ser el núcleo principal** de la aplicación, es decir:



Las APIs externas pueden ser utilizadas **para enriquecer** el proyecto (por ejemplo, obtener datos de ubicación, clima, divisas, etc.). Sin embargo, **el proyecto no puede depender exclusivamente de una API para funcionar**, ni puede basarse en simplemente mostrar lo que una API responde.

 Ejemplo no permitido:	 Ejemplos permitidos:
Un proyecto que simplemente llama a una API de recetas y muestra resultados en pantalla sin procesar ni almacenar nada, solo orquesta o hace de pasamanos con una interfaz.	Una app de seguimiento de actividades físicas que almacena datos del usuario, pero que usa una API externa para mostrar el clima del día o el precio de alimentos.

**NOTA:** Las API's son un tema que se manejará y profundizarán en Ingeniería de software 2, por lo que aunque pueden generar API's internas dentro del monolito, sería más trabajo para ustedes...

#### 5. Interfaz y Plataforma:

- El proyecto **debe ser una aplicación de escritorio con una interfaz visual (GUI)**.


Para este curso y tipo de proyecto, se busca que los estudiantes se enfoquen en la **lógica de negocio, el diseño limpio del código y la estructura del software**, no en problemas de configuración, despliegue ni ruteo web.

Quiero que par este curso se enfoquen en:

- Centrarse en la calidad del código y la arquitectura monolítica.
- Evitar distracciones innecesarias con tecnologías web que no son el foco del curso.
- Facilitar, integrar e inspeccionar directamente lógica de negocio, base de datos y pruebas sin capas adicionales.

**NOTA:** Se puede ejecutar en localhost la base de datos u otros recursos, pero la aplicación no debe exponer ni depender de un servidor web para interactuar con el usuario, es decir, por ahora solamente la BD debería correr en un localhost.

Semana	Fecha	Actividades	Requisitos del Proyecto
7	19-21 may	Clean Code Patrones de diseño (Singleton, Builder)	Tema del proyecto seleccionado
8	26-28 may	Más patrones (State, Observer) Static Testing	Inicio de Análisis de requerimientos y casos de uso
9	2-4 jun	FESTIVO Testing	
10	9-11 jun	TDD (Teoría) Clean Code (práctica)	Inicio del Hola Mundo funcional, diseño inicial de BD
11	16-18 jun	Avances proyecto TDD (práctica)	Comienzo de desarrollo activo

12	23-25 jun	FESTIVO o COMODÍN Introducción IA - 2	
13	30 jun - 2 jul	FESTIVO o COMODÍN Avances proyecto	
14	7-9 jul	Examen final avances proyecto	Pruebas finales, documentación
15	14-16 jul	Entrega final	 Entrega y presentación del proyecto