



Proyecto 1:

Casa Inteligente

PONDERACIÓN: 30

Horas Aproximadas: 30

Índice

1. Resumen Ejecutivo	3
2. Competencia que desarrollaremos	3
3. Objetivos del Aprendizaje	3
3.1 Objetivo General	3
3.2 Objetivos Específicos	3
4. Enunciado del Proyecto	5
4.1 Descripción del problema a resolver	5
4.2 Alcance del proyecto	5
4.4 Entregables	6
5. Metodología	8
6. Desarrollo de Habilidades Blandas	8
6.1 Proyectos en Grupo	8
6.2 Proyectos Individuales	9
7. Cronograma	13
8. Rúbrica de Calificación	13
8.1 Requisitos para optar a la calificación	13
8.2 Detalle de la Calificación	14
8.3 Valores	16
8.4 Comentarios Generales	16

1. Resumen Ejecutivo

El proyecto tiene como objetivo desarrollar un sistema de domótica funcional utilizando una Raspberry Pi, integrando sensores, actuadores y una aplicación web de control y monitoreo remoto, dentro del contexto de una vivienda inteligente para la residencial "Pinos Altos". Este sistema permitirá registrar y controlar diversos aspectos del hogar como temperatura, iluminación, alarmas, entre otros, mediante el uso de tecnologías IoT, utilizando el protocolo de comunicación MQTT y una base de datos en la nube (MongoDB). El proyecto también exige el desarrollo de un dashboard con visualización en tiempo real y una sección de control con autenticación de usuario. Además, se requiere una presentación profesional del prototipo físico y la entrega de manuales técnico y de usuario.

2. Competencia que desarrollaremos

Técnicas y tecnológicas:

- Manejo de entradas y salidas digitales con Raspberry Pi.
- Programación en Python con enfoque en estructuras de control y uso de librerías específicas.
- Implementación de protocolos de comunicación entre dispositivos (SPI, I2C, UART).
- Utilización de protocolo MQTT
- Desarrollo de aplicaciones web (React, Angular, Vue).
- Integración de base de datos en la nube (MongoDB) con almacenamiento de registros en formato JSON.

Analíticas y de resolución de problemas:

- Diseño de soluciones de automatización basadas en sensores y actuadores.
- Análisis y selección de gráficas representativas para datos de sensores.
- Diseño de interfaces gráficas intuitivas para el usuario final.

Trabajo colaborativo y gestión de proyectos:

- Trabajo en equipo (grupos de 5 integrantes).

- Uso de control de versiones (GitHub) con aportes registrados por todos los miembros.
- Entregas organizadas, con estructura y documentación profesional.

3. Objetivos del Aprendizaje

3.1 Objetivo General

El estudiante será capaz de diseñar y desarrollar una aplicación web que permita la automatización de dispositivos electrónicos en una vivienda, integrando el uso de hardware en Raspberry Pi con el lenguaje de Python

3.2 Objetivos Específicos

Al finalizar el proyecto, los estudiantes deberán ser capaces de:

1. Los estudiantes serán capaces de integrar sensores y actuadores a su proyecto, automatizando funciones como encender luces o controlar la temperatura en una vivienda.
2. Los estudiantes podrán diseñar y desarrollar interfaces gráficas que permitan encender o apagar dispositivos de forma remota y configurar automatizaciones sencillas.
3. Los estudiantes gestionarán su proyecto de manera ágil, organizando entregables, definiendo sprints y realizando revisiones constantes de su progreso.

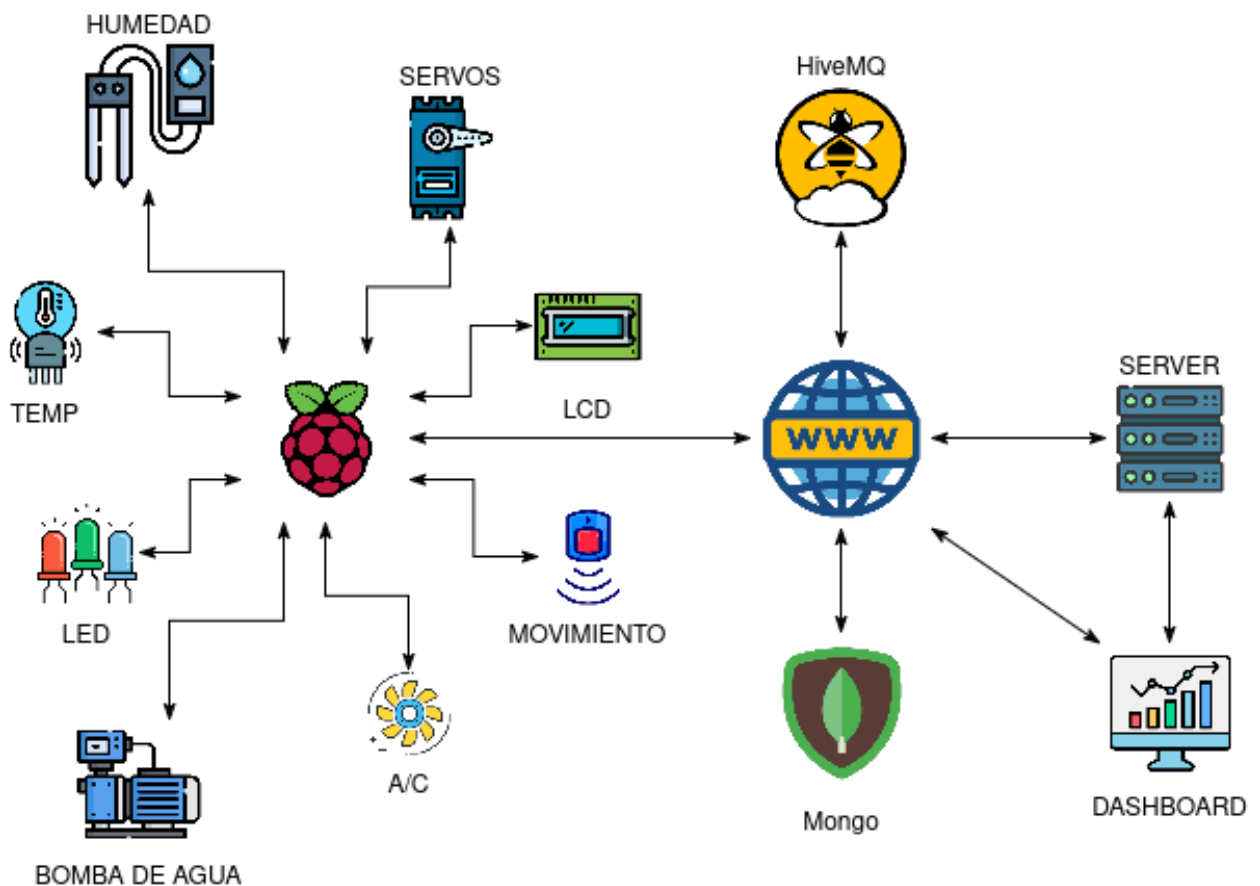
4. Enunciado del Proyecto

4.1 Descripción del problema a resolver

Su equipo de desarrollo ha sido contactado por la residencial “Pinos Altos”, ya que desean convertir algunas viviendas en lo que hoy conocemos como Smart Home. Los avances tecnológicos y la creciente demanda de automatización e interacción avanzada en los hogares han impulsado el desarrollo de sistemas de automatización residencial, también

conocidos como domótica. Estos sistemas aprovechan la disponibilidad cada vez mayor de componentes asequibles y permiten a los usuarios interactuar de manera más sofisticada con su entorno. En el ámbito de la investigación ambiental, este concepto se conoce como 'Smart Home' (hogar inteligente), que implica la instalación de sensores y actuadores en el hogar, así como la incorporación de hardware y software especializados que permiten el control remoto y la interacción deseada entre los diferentes dispositivos.

Arquitectura:



Iluminación:

El prototipo debe de contar con iluminación distribuida de la siguiente manera

- Un LED RGB que se podrá controlar desde la aplicación web

- Un LED RGB que funcionara como una alarma
- Tres LED blancos para iluminar algunas habitaciones.

La señal de encendido y apagado de las LEDS blancas y del LED RGB se enviará desde la aplicación web a través de la plataforma Hive MQ a través del canal “/illumination”. Del LED RGB se podrán hacer combinaciones de colores que se enviaran desde la página web. Cada vez que se encienda la iluminación de una habitación, deberá de mostrar en la pantalla LCD un mensaje indicando cual es la habitación que se ha encendido y deberá de realizar un registro en la base de datos, incluyendo, fecha, hora y una descripción de la habitación que se ha iluminado. De igual forma, deberá mostrar un mensaje cuando se apague la iluminación de la habitación indicando cuál ha sido.



Riego:

Se deberá de implementar un sistema de riego utilizando un sensor de humedad de tierra, cuando la humedad detectada sea menor al 95% se deberá de activar una bomba de agua pequeña para emular un sistema de riego a gran escala. Se deberá de almacenar los registros de las lecturas de humedad en la base de datos cada 7 minutos. De igual forma

cuando se active la bomba de agua se deberá de realizar un registro con la fecha y la hora en la que se activó la bomba y la lectura de humedad que ha detectado el sensor.

Ventilación:

Utilizando la combinación de sensor - actuador, se deberá de implementar un control de temperatura utilizando el sensor DHT11 o DHT22, que al tener lecturas de temperatura mayores a 22 °C deberá de accionar ventiladores que emulaban el accionar del Aire Acondicionado en el hogar, estos deberán de mantenerse activos hasta que la temperatura vuelva a disminuir por debajo de los 22 °C. Cada vez que se activen los ventiladores, se deberá de crear un par de registros nuevos en la base de datos en la nube, el primer registro debe de incluir fecha y hora en que se activó el sistema de ventilación y el segundo registro la fecha y hora en que se apagó. La lectura de la temperatura deberá de mostrarse constantemente en una pantalla LCD y deberán registrarse estas lecturas de temperatura cada 5 minutos en la base de datos.

De forma similar se deberá de implementar un sistema de alarma, en el cual cuando la temperatura sobrepase los 27 °C se deberá de activar un buzzer y utilizar LED RGB alternando entre los colores Rojo y Azul. Se realizará un registro en la base de datos con la fecha y hora en que se activó la alarma, la lectura de temperatura y se enviará una alerta que se deberá de ver en tiempo real en la página web a través del canal “/alerts”.

Movimiento:

Nuevamente, utilizando la combinación sensor - actuador, se le solicita que utilizando un sensor PIR para la parte exterior de la casa, emulando un patio o área verde principal, que al detectar un movimiento proceda con activar iluminación en el sitio. Cuando la iluminación se active en esta área NO deberá de mostrarse en la LCD, pero se deberá de hacer un registro en la base de datos donde se almacene la fecha y hora en que se detectó el movimiento únicamente cuando pase de un estado bajo a alto la lectura del sensor. Queda a discreción del grupo el tiempo pertinente que deberá transcurrir sin detección de movimiento para proceder con el apagado de la iluminación.

Entrada:

Se deberá implementar un portón automático, que se pueda accionar desde la app web, para esto, deberán de utilizar un servomotor, el cual al enviar la señal desde el dashboard a través de Hive MQ utilizando el canal “/entrance” para controlar la apertura del portón.

Una vez se ha ingresado por el portón, se deberá de enviar la señal para el cierre del portón. Por cada vez que se abra el portón, únicamente, se deberá de crear un registro en la base de datos indicando fecha y hora en que se realizó la acción. La acción de apertura y cierre deberá de indicarse en la pantalla LCD.

App Web:

Como parte de los sistemas IoT es importante poder tener un control y monitoreo de forma remota de los sensores y actuadores del sistema implementado. Por lo que deberán de desarrollar una aplicación web en el framework de su conveniencia (React, Angular, Vue, etc) o en HTML básico. Las secciones de esta aplicación web son las siguientes:

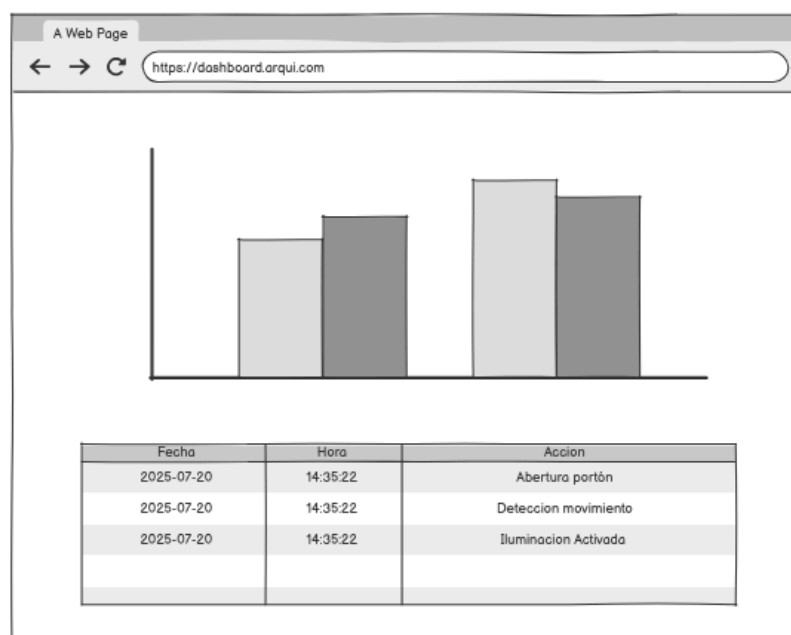
- **Login:**

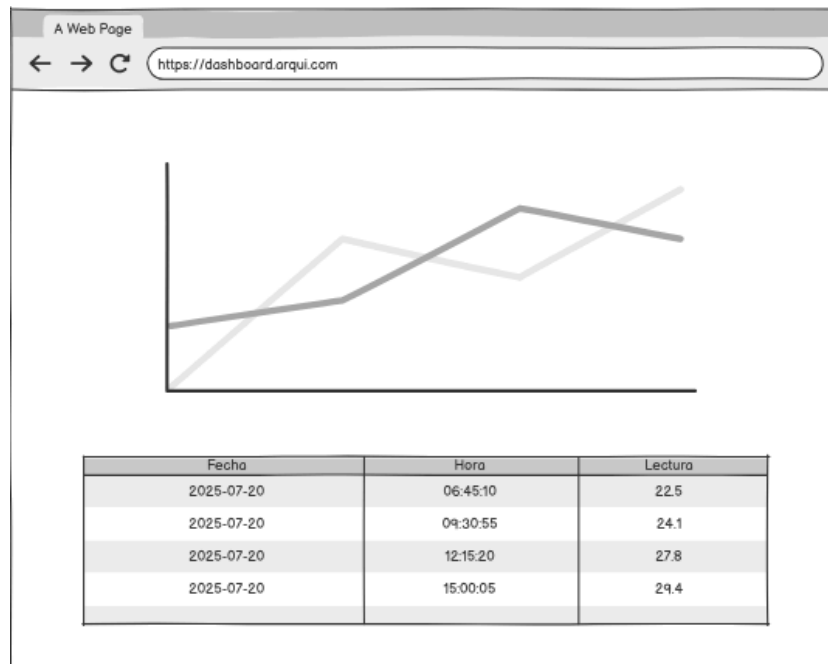
Para acceder a la aplicación web, deben de crear un login que permita acceder a las características de control y monitoreo de la aplicación. Se deberán de autenticar utilizando el siguiente usuario:

- User: **grupo#_seccion_proy1**
- Pass: **Discreción Del Grupo**

- **Dashboard:**

Deberán de crear un dashboard interactivo en donde se pueda monitorear las lecturas de los sensores y estados de los actuadores. Por lo que los registros almacenados en la base de datos en la nube, específicamente en MongoDB, deberán de visualizarse en este apartado. Se especifican sugerencias del contenido del dashboard a continuación.





- **Restricciones Dashboard:**

- Deberán contar con lecturas de mínimo 3 días
- Deben de almacenar la fecha y hora de los registros para agrupar los datos por fecha o por hora en las gráficas.
- Para los actuadores se debe usar una gráfica de barras y para las lecturas de temperatura se debe de utilizar una gráfica de líneas.
- Deben de mostrar los registros almacenados tanto en gráficas como en tablas para su respectivo filtrado.

- **Control**

Se deberá de poder realizar control remoto de los actuadores mencionados anteriormente a través de la aplicación web, es decir se podrá controlar la iluminación y la entrada desde la aplicación web, mediante el servicio de HiveMQ.

NOTA: Se tomará en cuenta el diseño atractivo de la aplicación web así como su intuitividad para la facilidad de uso para cualquier usuario. Por lo que pueden utilizar cualquier librería de diseño. Ya que la aplicación se correrá de forma local en una PC de alguno de los integrantes del grupo.

Servidor:

Se implementara un servidor que será el encargado de comunicar la página web con la base de datos a través de una API. El lenguaje a utilizar quedará a discreción del grupo.

Materiales Sugeridos:

- Raspberry Pi 3 o 4
- Ventiladores o Motores de 5V
- LCD o OLED
- Leds
- Resistencias
- Conversor de 4 u 8 canales de 3.3V a 5V
- Flip Flops tipo D
- Multiplexores
- Protoboard
- Jumpers
- Cargadores
- Transistores
- Servomotores
- Sensor PIR o ULTRASÓNICO

4.2 Alcance del proyecto

Incluye:

- *Desarrollo de un prototipo físico de Smart Home funcional.*
- *Implementación de sensores y actuadores (temperatura, iluminación, acceso).*
- *Integración de una API para lectura de registros.*
- *Aplicación web con:*
 - *Dashboard de monitoreo (con gráficas y tablas).*
 - *Control remoto autenticado.*
- *Base de datos MongoDB con almacenamiento de lecturas por al menos 3 días.*
- *Documentación técnica y manual de usuario.*
- *Encapsulado profesional del sistema físico.*

- *Entrega en plataforma UEDI con repositorio estructurado.*

No incluye:

- *Implementaciones en servicios de hosting productivo (la app web corre localmente).*
- *Desarrollo para múltiples viviendas (solo un hogar prototipo).*
- *Garantías de escalabilidad del sistema a producción industrial.*

4.3 Entregables

- **Prototipo:** Desarrollo de una versión funcional del proyecto, que permite visualizar su funcionamiento básico a escala.
- **Informe técnico:** Un documento detallado que describe el proceso de desarrollo, decisiones tomadas por el equipo de forma justificada, costos del prototipo, resolución de problemas, especificación de roles del equipo, especificaciones del equipo y tecnologías utilizadas, servicios agregados al proyecto y diagramas de conexión realizados en proteus.
- **Código Fuente:** Todo el código fuente utilizado en la construcción del prototipo, debidamente identificado en un repositorio.

5. Metodología

Incremental, debido a que se deben de construir diferentes capas del prototipo por partes, iniciando con la parte electrónica, luego a medida que se va avanzando se agregan las demás funcionalidades de software.

Ejemplo de Fases:

1. Lectura de sensores y prototipo físico básico.
2. Integración con servicios Hive MQ y Mongo, Desarrollo de Frontend y Backend.
3. Integración software con prototipo físico electrónico.
4. Recopilación de datos y corrección de errores.
5. Pruebas finales y elaboración de documentación

6. Desarrollo de Habilidades Blandas

Para complementar el desarrollo técnico, esta sección se centra en las habilidades blandas que los estudiantes deberán mejorar a lo largo del proyecto, como la comunicación, el liderazgo, y la colaboración en equipo.

6.1 Proyectos en Grupo

El trabajo en equipo es clave para el éxito de cualquier proyecto. Los estudiantes deben colaborar, asignando roles específicos (líder de proyecto, desarrollador, diseñador, etc.) para organizar y distribuir tareas de manera eficiente. A través de herramientas como Trello, Google Docs o GitHub, los equipos gestionarán sus avances y compartirán información de manera eficaz.

Además, se fomenta la toma de decisiones conjunta y la resolución de problemas en grupo, desarrollando habilidades como la comunicación, el liderazgo y la negociación. Los grupos recibirán retroalimentación constante para mejorar sus soluciones y adaptarse a los desafíos que surjan.

6.1.1 Trabajo en Equipo

Cada grupo debe trabajar en colaboración, asignando roles específicos a cada miembro (líder de proyecto, desarrollador, diseñador, etc.). Los estudiantes deben coordinarse mediante herramientas de gestión como Trello o GitHub. Se deberá de establecer una organización de grupo en donde se defina un rol en específico para cada integrante del grupo.

6.1.2 Comunicación Efectiva

Los grupos deben realizar presentaciones periódicas ante sus compañeros y el profesor, compartiendo los avances y problemas encontrados, y recibiendo retroalimentación para mejorar sus soluciones.

6.1.3 Resolución de Conflictos

A lo largo del proyecto, los estudiantes pueden enfrentar desafíos o desacuerdos. Deberán aplicar técnicas de resolución de conflictos para mantener la armonía y la productividad en el equipo.

7. Cronograma

Tipo	Fecha Inicio	Fecha Fin
Asignación de Proyecto	Semana 3	Semana 3
Elaboración	Semana 3	Semana 8
Calificación	Semana 8	Semana 8

8. Rúbrica de Calificación

8.1 Requisitos para optar a la calificación

Tema	Descripción	Cumple (Si/No)
Cumplimiento de la tecnología establecida	El desarrollo debe haberse realizado en el lenguaje de programación especificado (Python), en la raspberry pi.	
Gestión y entregas del proyecto	Deben de haber participación de todos los integrantes evidenciado en sus entregas en el repositorio.	
Documentación obligatoria	Se debe entregar documentación técnica completa, incluyendo manual de usuario.	
Pruebas y funcionalidad mínima	El sistema debe ser funcional. Debe haber evidencia de pruebas realizadas y reporte de resultados.	

8.2 Detalle de la Calificación

Descripción de Ponderación	Valor	Observación	Punteo
Sensores y Raspberry Pi	25		
Conexión correcta Led RGB 1	1.5		
Conexión correcta Led RGB 2 (alarma)	1		
Conexión correcta Led 1	0.5		
Conexión correcta Led 2	0.5		
Conexión correcta Led 3	0.5		
Conexión correcta sensor de temperatura	1		
Conexión correcta ventilador	1		
Conexión correcta puerta principal	2		
Conexión correcta sensor de movimiento	2		
Conexión correcta bomba de agua	3		
Conexión correcta sensor humedad	3		
Conexión correcta pantalla LCD	4		
Encapsulamiento del prototipo	5		
Programa Python	20		
Encender/apagar Led RGB 1	1.5		
Activación RGB 2 (alarma)	1		
Encender/apagar Led 1	0.5		
Encender/apagar Led 2	0.5		
Encender/apagar Led 3	0.5		
Lectura correcta sensor de temperatura	1		
Encender/apagar ventilador	1		
Accionar puerta principal	2		
Lectura correcta sensor de movimiento	2		
Lectura correcta sensor de humedad	3		
Activación bomba de agua	2		
Funcionamiento correcto pantalla LCD	5		
Mongo DB	10		
Almacenamiento LED RGB Alarma	0.5		
Almacenamiento LED 1	0.5		
Almacenamiento LED 2	0.5		
Almacenamiento LED 3	0.5		
Almacenamiento ventilador	0.5		
Almacenamiento temperatura	1		
Almacenamiento sensor de proximidad	0.5		

Almacenamiento bomba de agua	0.5		
Almacenamiento puerta principal	0.5		
Formato con fecha y hora	2.5		
Almacenamiento mínimo de 3 días de antelación	2.5		
API	10		
API funcional	4		
Consulta correcta de datos desde MongoDB	3		
Manejo de rutas específicas para cada tipo de dato o sensor	3		
Dashboard	15		
Visualización clara de gráficas (líneas para temperatura, barras para eventos)	5		
Tablas con filtros por hora o fecha	5		
Control remoto vía MQTT	2		
Login funcional	1		
Diseño atractivo e intuitivo	2		
MQTT	10		
Encender/apagar Led RGB Rojo	1		
Encender/apagar Led RGB Verde	1		
Encender/apagar Led RGB Azul	1		
Activación RGB 2 (alarma)	1		
Encender/apagar Led 1	1		
Encender/apagar Led 2	1		
Encender/apagar Led 3	1		
Encender/apagar ventilador	1		
Accionar puerta principal	1		
Activación bomba de agua	1		
Documentación	5		
Manual Técnico	2.5		
Manual de Usuario	2.5		
Preguntas	5		
Pregunta 1	1		
Pregunta 2	1		
Pregunta 3	1		
Pregunta 4	1		
Pregunta 5	1		

8.3 Valores

En el desarrollo del proyecto, se espera que cada estudiante demuestre honestidad académica y profesionalismo. Por lo tanto, se establecen los siguientes principios:

1. Originalidad del Trabajo

- Cada estudiante o equipo debe desarrollar su propio código y/o documentación, aplicando los conocimientos adquiridos en el curso.

2. Prohibición de Copias y Plagio

- Si se detecta la copia total o parcial del código, documentación o cualquier otro entregable, la calificación será de **0 puntos**.
- Esto incluye la reproducción de código entre compañeros, la reutilización de proyectos de semestres anteriores o el uso de código externo sin la debida referencia.

3. Uso Responsable de Recursos Externos

- El uso de bibliotecas, frameworks y ejemplos de código externos está permitido, siempre y cuando se referencian correctamente y se comprendan plenamente. (Consultar con el catedrático su política)

4. Revisión y Detección de Plagio

- Se podrán utilizar herramientas automatizadas y revisiones manuales para identificar similitudes en los proyectos.
- En caso de sospecha, el estudiante deberá justificar su código y demostrar su desarrollo individual o en equipo. Si este extremo no es comprobable la calificación será de **0 puntos**.

Al detectarse estos aspectos se informará al catedrático del curso quien realizará las acciones que considere oportunas.

8.4 Comentarios Generales
