

1. Definición del Problema y Contexto

- Breve descripción de la problemática industrial o social identificada.
- Justificación de la relevancia: ¿Por qué merece la pena transformarla mediante un gemelo digital o sistema inteligente?
- Contexto específico (hospital, smart home, piscifactoría...)

2. Estado del Arte y Antecedentes

- Revisión crítica de al menos 3-5 soluciones previas o aproximaciones existentes (artículos, webs, proyectos académicos).
- Identificación de tecnologías empleadas, límites y lecciones aprendidas de otros proyectos.
- Referencias bibliográficas según el sector.

3. Arquitectura Propuesta

- Diagrama global: Incluyendo todos los módulos principales, aunque todos sean software/simulación.
- Componentes clave:
 - Sensores virtuales/datos sintéticos (qué se mide y por qué).
 - Plataforma de integración y procesamiento (e.g., ThingsBoard, Python, Node-RED, SimPy/AnyLogic/FlexSim, herramientas cloud o open source).
 - Gemelo digital/modelo de simulación (físico, bioenergético, flujo, RL, etc.).
 - Análisis y visualización (dashboards, Power BI, Grafana, visualización web o vídeo).
- Flujos de datos entre componentes.
- Señalar interoperabilidad/estándares empleados (MQTT, HTTP, REST API...)

4. Implementación Detallada

- Descripción técnica de cada módulo del prototipo:
 - Cómo se generan y procesan los datos (simulación, datasets, scripts de Python...).
 - Qué variables entran/salen, frecuencia, lógica de simulación.
 - Cómo funciona el gemelo digital (método, modelo, integración).
 - Componente(s) de inteligencia artificial, si aplica (ej: RL, CLASIFICACIÓN, predicción).

- Integración/visualización en dashboard.
- Código relevante (fragmentos esenciales, github/repo/documentación como evidencias, scripts/entorno de simulación reproducible).

5. Resultados y Validación

- Pruebas realizadas: escenarios base vs optimizados.
- KPIs/métricas principales evaluadas (tiempo medio de espera, ahorro energético, reducción de fallos/mortalidad, eficiencia...).
- Visualización de resultados: gráficas, capturas de dashboard, vídeos de simulación.
- Comparativa con la situación actual (si hubiera datos de referencia).
- Limitaciones y lecciones aprendidas durante el desarrollo.

6. Conclusiones y Futuras Mejoras

- Valor generado por el prototipo (en cifras estimadas, si es posible).
- Posibles avances futuros: integración con datos reales, ampliación de escenarios, nuevas tecnologías.
- Transferibilidad a otros ámbitos/sectores.

7. Material de Apoyo y Presentación

- Diapositivas para exposición clara (máx. 8 minutos, todos los miembros deben participar).
- Documentación organizada en carpeta/repo, incluyendo manual de uso o README reproducible.

Recomendaciones

- Simular sensores/actuadores con scripts o plataformas (ejemplo: Python random generators, miniprogramas, dashboards online).
- Utilizar demos online de plataformas IoT (ThingsBoard demo, Node-RED online, simuladores de AnyLogic...).
- En proyectos de IA/RL, documentar ambiente virtual y comportamiento del agente en escenarios de entrenamiento/simulación.
- Incluir “pantallazos” o grabaciones de flujo de información, simulación y dashboards para la presentación.

- Destacar la integración conceptual (aunque sea virtual), mostrando el flujo de información desde entrada de datos hasta resultado final.
- Resaltar cumplimiento de resultados de aprendizaje y competencias de la asignatura en cada fase del proyecto.