

Actividad Final Propuesta Docente

Elabora una propuesta docente alineada con los resultados de aprendizaje de tu asignatura, que contenga una actividad significativa para el perfil profesional. La actividad deberá promover el aprendizaje profundo y la autorregulación del estudiantado siguiendo los modelos de Pintrich y Winne-Hadwin mostrados en el taller.

1. Rellena la ficha con los datos de la actividad.

Titulación	Grado en Tecnologías Interactivas (GTI)
Asignatura	Programación 2
Curso	2º
Título de la Actividad	Encuentra el agente secreto y completa la misión
Resultado de aprendizaje de la Asignatura que permite evaluar/trabajar la actividad	RA1: Diseñar e implementar programas en JavaScript que gestionen la lectura y escritura de ficheros y el flujo asíncrono mediante callbacks, utilizando librerías fundamentales de Node.js. RA2: Desarrollar algoritmos de procesamiento de datos (normalización, filtrado y agregación) justificando las decisiones de diseño adoptadas.
Objetivos de la Actividad	Al finalizar la actividad, el estudiante deberá ser capaz de: <ul style="list-style-type: none">• Procesar ficheros JSON mediante una secuencia de operaciones de lectura, transformación y escritura usando callbacks.• Aplicar correctamente funciones de orden superior (map, filter, reduce) para resolver un problema algorítmico concreto.• Diseñar funciones modulares y reutilizables que faciliten la escalabilidad y el mantenimiento del código.• Analizar errores detectados mediante un oráculo o pruebas automáticas y corregirlos de forma razonada.• Reflexionar sobre el impacto de las decisiones de diseño en la legibilidad y mantenibilidad del código.
Planificación de la Actividad	Duración total: 2 semanas Estructura por fases: <ul style="list-style-type: none">• Fase 0: Activación metacognitiva y planificación (planificacion.md).• Fase 1: Limpieza y normalización de datos con map y filter.• Fase 2: Algoritmia avanzada con reduce (búsqueda del punto de corte).• Fase 3: Búsqueda del agente secreto mediante composición funcional.• Fase 4: Reflexión final y autorregulación (reflexion_final.md). Feedback: Validación automática mediante oráculo tras cada fase. Modalidad: Trabajo individual, autónomo y progresivo.

Dimensiones de autorregulación que estás trabajando (Pintrich, 1995)	<p>Cognición:</p> <ul style="list-style-type: none"> • Planificación de la solución. • Monitorización de la comprensión durante la implementación. • Evaluación de decisiones algorítmicas. <p>Motivación y afecto:</p> <ul style="list-style-type: none"> • Metas de dominio (resolver el problema correctamente, no solo “que funcione”). • Refuerzo de la autoeficacia mediante feedback inmediato del oráculo. <p>Comportamiento:</p> <ul style="list-style-type: none"> • Gestión del tiempo (trabajo por fases). • Búsqueda de ayuda regulada (oráculo, pruebas, reflexión).
Describe como has integrado el modelo COPES en tu actividad.	<p>Condiciones</p> <ul style="list-style-type: none"> • Enunciado estructurado por fases. • Restricciones técnicas claras (callbacks, librerías permitidas). • Oráculo como fuente de feedback externo. <p>Operaciones</p> <ul style="list-style-type: none"> • Planificación previa (Fase 0). • Aplicación de estrategias cognitivas: descomposición, abstracción, composición funcional. • Uso deliberado de map, filter y reduce. <p>Productos</p> <ul style="list-style-type: none"> • Archivos phase1.json, phase2.json, phase3.json. • Código fuente modular. • Documentos de planificación y reflexión. <p>Evaluaciones</p> <ul style="list-style-type: none"> • Validación automática del oráculo. • Autoevaluación mediante checklist y reflexión final. <p>Estándares</p> <ul style="list-style-type: none"> • Corrección funcional. • Claridad y modularidad del código. • Justificación razonada de decisiones. <p>Fases del modelo</p> <ol style="list-style-type: none"> 1. Definición de la tarea: comprensión global del flujo de datos (Fase 0). 2. Establecimiento de metas y planificación: diseño de funciones y callbacks. 3. Ejecución: implementación progresiva por fases con feedback continuo. 4. Adaptación: ajustes tras errores detectados y reflexión final para futuras tareas.
Los estudiantes ¿tendrán que usar IAGen para resolver la actividad?	<p>No.</p> <p>El uso de IA generativa no es obligatorio, aunque está permitido como apoyo al aprendizaje bajo las condiciones indicadas en el enunciado.</p>
Si los alumnos tienen tienen que usar la IAGen	

Describe como has integrado el modelo didáctico de 6 pasos	No aplica, ya que el uso de IA generativa es opcional y no forma parte del diseño didáctico obligatorio de la actividad.
--	--

- Elabora una infografía (puedes usar NotebookLM) para presentar la actividad al resto de compañeros en la sesión del 14 de Abril.

