

Parte Didáctica

Módulo: Programación – 1.º DAM

Actividad: “Del problema a la pantalla”

2.1 ¿Qué supuesto queremos trabajar?

 Simular un proceso completo de desarrollo algorítmico:

- Planteamiento de un problema cotidiano
- Diseño de la solución en pseudocódigo y diagramas
- Codificación en lenguaje real (Python)
- Representación visual del algoritmo mediante un prototipo de interfaz

2.2 Contextualización del alumnado

 Alumnado de 1.º DAM

- En proceso de adquirir competencias en programación estructurada
- Grado de heterogeneidad medio-alto
- Motivación vinculada a la creación de soluciones reales y visuales

2.3 Conocimientos previos requeridos






- ✓ Uso de condicionales (`if` , `else`)
- ✓ Bucles (`for` , `while`)
- ✓ Entrada/salida en consola
- ✓ Funciones simples
- ✓ Nociones básicas de lógica algorítmica

2.4 Objetivos de aprendizaje

 El alumnado será capaz de:

- Analizar problemas y plantear soluciones paso a paso
- Representar algoritmos mediante pseudocódigo y diagramas
- Traducir algoritmos a código funcional en Python
- Diseñar interfaces visuales que reflejen el comportamiento lógico del algoritmo

2.5 Metodología

-  Enfoque práctico y por fases
-  Aprendizaje basado en retos cotidianos
-  Evaluación continua y formativa
-  Trabajo individual con momentos colaborativos
-  Retroalimentación inmediata

2.6 Material didáctico (DUA)

- Plantilla base de pseudocódigo y diagramas
- Fichas guía con ejemplos visuales
- Acceso a entorno Python (Thonny, Replit...)
- Introducción guiada a Figma (versión gratuita)
- Adaptación textual y visual del enunciado

2.7 Secuencia de acciones formativas

1. Presentación de varios problemas cotidianos
2. Elección del problema y análisis de entradas/salidas
3. Representación de la solución en:
 - Pseudocódigo
 - Diagrama de flujo o Nassi-Shneiderman
4. Codificación en Python
5. Prototipado del flujo en Figma
6. Presentación y revisión cruzada entre compañeros

2.8 Actividad principal

“Del problema a la pantalla”

Fase 1: Planteamiento del problema

- Ejemplos: sistema de descuentos, gestión de reservas, turnos, votaciones...

Fase 2: Diseño de la solución

- Pseudocódigo
- Diagrama de flujo o Nassi-Shneiderman

Fase 3: Implementación en Python

- Código funcional a partir del diseño lógico

Fase 4: Prototipado visual

- Interfaz en Figma que refleje pantallas, botones, lógica de interacción

2.9 Evaluación: instrumentos y criterios

Criterios:

- Análisis correcto del problema
- Coherencia entre pseudocódigo, diagrama y código
- Funcionalidad del programa
- Relación entre lógica y flujo visual
- Claridad y creatividad del prototipo

Instrumentos:

- Lista de control por fases
- Rúbrica detallada
- Observación directa + entrega final

2.10 Inclusión y atención a la diversidad

- Plantillas y ejemplos adaptados
- Uso flexible del entorno de desarrollo
- Opción de entregar por fases o por pareja
- Prototipado guiado con recursos visuales
- Feedback personalizado

2.11 Actividades de ampliación


🌟 Para alumnado avanzado:

- Integrar entrada/salida gráfica (Tkinter, PySimpleGUI)
- Añadir persistencia básica de datos
- Simular casos de error y validación
- Crear documentación técnica y vídeo demo

Cierre

 La actividad conecta todas las fases del desarrollo de una aplicación:

- Problema → Análisis → Lógica → Código → Interfaz

 Reproduce el ciclo completo de pensamiento computacional y diseño orientado al usuario.