Oposiciones cuerpo de secundaria.

Esquemas dos páginas sobre temario oposición profesorado Secundaria.

Especialidad informática

Autor: Sergi García Barea

Actualizado Mayo 2025

Licencia

Índice

Introducción

Este documento recoge una serie de **esquemas sintéticos del temario oficial para las oposiciones al cuerpo de profesorado de Secundaria, especialidad Informática**, con el objetivo de ofrecer una herramienta de estudio clara, útil y eficaz. Cada esquema está diseñado para ocupar como máximo **cuatro páginas**, facilitando así su consulta rápida, comprensión global y memorización eficaz.

Para el buen docente

Pero estos esquemas **no son solo para superar una oposición**. Están pensados para ayudarnos a **ser mejores docentes**, personas que entienden la complejidad técnica de su materia, pero también su dimensión educativa, social y ética. Ser docente es una tarea de gran responsabilidad que trasciende un examen: **enseñamos a través de lo que sabemos, pero también a través de lo que somos**.

Por eso, si has llegado hasta aquí, te pido algo importante: lleva contigo el compromiso de ser un buen docente más allá de la oposición. Utiliza estos materiales como base, sí, pero hazlos crecer con tu experiencia, tus reflexiones y tu vocación. Que enseñar sea una decisión consciente, diaria, y no un trámite. Que lo que prepares hoy, lo apliques con compromiso durante toda tu carrera docente, pensando siempre en lo mejor para tu alumnado.

¿Para qué prueba están adaptados estos esquemas?

Estos esquemas están específicamente adaptados para la **prueba de exposición oral del procedimiento selectivo regulado por la ORDEN 1/2025, de 28 de enero**, de la Conselleria de Educación, Cultura, Universidades y Empleo de la Comunitat Valenciana, que establece lo siguiente:

"La exposición tendrá dos partes: la primera versará sobre los aspectos científicos del tema; en la segunda se deberá hacer referencia a la relación del tema con el currículum oficial actualmente vigente en el presente curso escolar en la Comunitat Valenciana, y desarrollará un aspecto didáctico de este aplicado a un determinado nivel previamente establecido por la persona aspirante. Finalizada la exposición, el tribunal podrá realizar un debate con la persona candidata sobre el contenido de su intervención."

No obstante, estos materiales pueden ser también útiles para preparar **otras modalidades de oposición** (como ingreso por estabilización o pruebas de adquisición de especialidades), así como para otras especialidades cercanas, especialmente **la de Sistemas y Aplicaciones Informáticas**, ya que comparten gran parte del temario técnico

Tema 1: Representación y comunicación de la información

1.1 Introducción

- Definición: transformación de fenómenos del mundo real en estructuras digitales binarias.
- Fundamento para todo procesamiento informático: desde el código hasta el hardware.

1.2 Sistemas de Numeración

- Base, dígitos, sistema posicional.
- Sistemas: binario, octal, hexadecimal y decimal.
- Conversión entre sistemas para debugging, direccionamiento y arquitectura.

1.3 Representación de Datos

- **Enteros:**
  - Códigos: signo+magnitud, CA1 y CA2 (uso de CA2 por su simplicidad en hardware).
- **Punto flotante (IEEE 754):**
  - Precisión simple (32 bits) y doble (64 bits), compuestas por signo (1 bit), exponente (8/11 bits) y mantisa (23/52 bits).
  - Se normaliza desplazando el punto binario para que quede en la forma 1.xxxxx, permitiendo maximizar la precisión y garantizar una representación única.
- **Texto:**
  - ASCII (7 bits, limitado), 8 bits extendido y Unicode (UTF-8/16/32) para múltiples idiomas y emoji.
- **Imágenes:**
  - Representación como matriz de píxeles con RGB + canal alfa.
  - Formatos: BMP, PNG (sin pérdida), JPEG (con pérdida).
- **Vídeo:**
  - Secuencia de imágenes y audio. Compresión intraframe e interframe.
  - Códecs: H.264, H.265 (HEVC), AV1 (compresión espacial e interframe).
- **Audio:**
  - Muestreo (44,1 kHz CD / 48 kHz vídeo); cuantificación (16/24 bits).
  - Formatos: WAV/FLAC (sin pérdida), MP3/AAC (con pérdida).

1.4 Lógica y Operaciones Binarias

- Aritmética: suma, resta, multiplicación, división en base 2.
- Lógica digital: puertas AND, OR, NOT, XOR.
- Aplicaciones: ALU, circuitos combinacionales.

1.5 Detección y Corrección de Errores

- Bit de paridad (detección simple).
- CRC (Cyclic Redundancy Check).
- Código de Hamming (corrige 1 bit; MEM ECC).
- Reed-Solomon (múltiples errores; CDs, RAID, QR).

1.6 Representación en Big Data y Nube

- Formatos JSON/BSON, Avro, Protobuf, Parquet, ORC.
- Integración en pipelines cloud, microservicios y análisis masivo (Spark, Hadoop).

1.7 Comunicación Digital

- Modelo Shannon-Weaver: emisor, codificador, canal (ruido), decodificador receptor + (Compresión y cifrado)
- Señales digitales vs analógicas.
- Protocolos: TCP/IP, UDP/IP, Ethernet, WebRTC.

1.8 Seguridad

- Hashing: SHA-256 (integridad), bcrypt/Argon2 (contraseñas).
- Cifrado:
  - Simétrico: AES.
  - Asimétrico: RSA, ECC.
  - Cifrado simétrico pero compartiendo la clave con cifrado Asimétrico (SSL)
  - Aplicaciones: HTTPS, VPN, BitLocker, TLS.

1.9 Compresión

- Sin pérdida: Huffman, LZW (ZIP, PNG).
- Con pérdida: JPEG, MP3, H.264 (optimizadas para multimedia).

1.10 Conclusión

- La correcta representación y manipulación de datos es básica en informática.
- Aplica desde circuitos y sistemas embebidos hasta servicios cloud y Big Data.

2. PARTE DIDÁCTICA - 1.º DAM – Módulo: Programación

🗂️ Unidad didáctica: "Estructuras repetitivas aplicadas a la representación y tratamiento de información"

2.1 Contextualización didáctica

- Nivel: 1.º de Ciclo Formativo de Grado Superior – Desarrollo de Aplicaciones Multiplataforma (DAM)
- Módulo profesional: Programación
- Currículo aplicable: Real Decreto 450/2010 y Decreto 48/2011 (Comunitat Valenciana)
- **Justificación:** Las estructuras repetitivas permiten automatizar tareas, manipular información y modelar procesos fundamentales en programación profesional.

2.2 Objetivos de aprendizaje

- Aplicar bucles (for, while, do-while) en la codificación de algoritmos.
- Manipular datos numéricos y textuales mediante estructuras de control repetitivas.
- Simular procesos de codificación, verificación y transmisión de datos.

2.3 Metodología y principios pedagógicos

- Metodología activa: basada en tareas prácticas y resolución de problemas reales.
- Progresión: ejercicios con dificultad creciente y trabajo individual seguido de refactorización colaborativa.
- Recursos utilizados: IDE Java (NetBeans, VS Code), pseudocódigo, diagramas de flujo, vídeos explicativos.
- Estrategias metodológicas: trabajo por parejas con roles diferenciados, flipped classroom, revisión entre iguales.

2.4 Inclusión y atención a la diversidad (niveles III y IV):

- Código base parcial con comentarios orientativos.
- Retroalimentación individualizada durante el proceso.

2.5 Aplicación del Diseño Universal para el Aprendizaje (DUA):

- Múltiples formas de representación (textual, visual, audiovisual).
- Variedad de formas de expresión del aprendizaje (código funcional, exposición oral, demos).
- Participación equitativa mediante ajustes de complejidad y agrupaciones heterogéneas.

2.6 Actividad principal: "Procesando datos binarios con bucles"

- Conversión manual de números decimales a binario, octal y hexadecimal mediante bucles.
- Codificación de texto carácter a carácter en binario utilizando la tabla ASCII.
- Cálculo de bit de paridad mediante conteo de unos en cadenas binarias.
- Simulación de un canal de transmisión con errores aleatorios y aplicación del Código de Hamming.
- Verificación de integridad de cadenas binarias utilizando un algoritmo de hash simplificado (XOR).

2.7 Evaluación

- Criterios de evaluación: uso correcto y eficiente de bucles, lógica de control adecuada, limpieza del código, comprensión de los procesos implicados.
- Instrumentos de evaluación: rúbricas detalladas, revisión entre compañeros, evaluación continua con entregas parciales.
- Resultados esperados: desarrollo de programas funcionales que evidencien el dominio de las estructuras repetitivas aplicadas a tareas reales de representación y transmisión de datos.

2.8 Conclusión didáctica

Esta unidad permite al alumnado integrar conocimientos fundamentales de programación y aplicarlos en situaciones prácticas. Fomenta el pensamiento algorítmico, el desarrollo de habilidades técnicas y competencias transversales, incrementando la motivación y la autonomía. Además, establece conexiones claras entre la teoría informática y su aplicación profesional, desarrollando un aprendizaje significativo.

Tema 2: Elementos funcionales de un ordenador digital. Arquitectura.

1. INTRODUCCIÓN
2. ELEMENTOS FUNCIONALES
2.1. Unidad Central de Proceso (CPU - Central Processing Unit)
Encargada de ejecutar instrucciones del programa.

- **Registros**: PC, IR, MAR, MDR, FLAGS.
- **ALU / FPU**: operaciones lógicas y en coma flotante.
- **Unidad de Control**: cableada (rápida) o microprogramada (flexible).

2.2. Memoria principal
Memoria de acceso rápido que almacena temporalmente datos e instrucciones.

- **RAM (Random Access Memory)**:
  - **DRAM (Dynamic RAM)**: económica, necesita refresco constante.
  - **SRAM (Static RAM)**: más rápida, usada en cachés.
- **Jerarquía de memoria**: estructura escalonada que optimiza acceso:
  - Registros > Caché (L1, L2, L3) > RAM > SSD/HDD.
- Afecta directamente al **rendimiento**: menor latencia en niveles superiores.

2.3. Subsistema de Entrada/Salida (E/S)
Permite la interacción del procesador con dispositivos externos.

- **Dispositivos periféricos**: teclado, ratón, impresora, disco, red.
- **Modos de transferencia**:
  - **Polling**: la CPU consulta activamente si hay datos disponibles.
  - **Interrupciones**: el periférico avisa al procesador cuando necesita atención.
  - **DMA (Direct Memory Access)**: transfiere datos directamente sin CPU.

2.4. Sistema de buses
Canales físicos que interconectan los componentes del sistema.

- **Tipos**: **Bus de datos** (transmite información)**, bus de direcciones** (localiza posiciones de memoria) y **bus de control** (gestiona operaciones como lectura o interrupciones).
- **Temporización**: puede ser **síncrona** (con reloj compartido) o **asíncrona** (mediante señales independientes, más flexible).

3. MODELOS DE ARQUITECTURA
3.1. Von Neumann

- Memoria compartida para instrucciones y datos.
- Problema: **cuello de botella** en el acceso a memoria.

3.2. Harvard

- Memoria separada para datos e instrucciones.
- Permite acceso paralelo, más eficiente.

4. TAXONOMÍA DE FLYNN
Clasificación de arquitecturas según número de flujos de instrucciones y datos:

- **SISD (Single Instruction, Single Data)**: tradicional, una instrucción opera sobre un dato.
- **SIMD (Single Instruction, Multiple Data)**: una instrucción actúa sobre múltiples datos (p. ej., GPU).
- **MISD (Multiple Instruction, Single Data)**: redundante, escasa utilidad práctica.
- **MIMD (Multiple Instruction, Multiple Data)**: múltiples procesadores ejecutan múltiples instrucciones, típico en sistemas multinúcleo.

5. MEMORIAS: TIPOS Y EVOLUCIÓN

- **ROM (Read-Only Memory)**: no volátil, incluye BIOS/UEFI.
  - **EEPROM**: puede reprogramarse eléctricamente.
- **Flash**: memoria no volátil usada en SSD y dispositivos móviles.
- **Tendencias actuales**:
  - **HBM (High Bandwidth Memory)**: gran ancho de banda, muy cercana al procesador.
  - **GDDR6**: memoria gráfica usada en GPUs.
  - **Optane**: tecnología de Intel basada en memoria persistente de alta velocidad.
  - **SoC (System on Chip)**: integración total en un único chip, común en móviles.

6. CICLO DE INSTRUCCIÓN
Etapas secuenciales que sigue la CPU para ejecutar una instrucción:

1. **Fetch**: se lee la instrucción desde memoria.
2. **Decode**: se interpreta la instrucción.
3. **Execute**: se realiza la operación.
4. **Memory**: acceso a memoria si es necesario.
5. **Write-back**: los resultados se guardan.

Técnicas de optimización:

- **Pipeline**: ejecución en paralelo de etapas.
- **Superescalaridad**: ejecución de múltiples instrucciones por ciclo.
- **Out-of-Order Execution**: reordenamiento dinámico para mejorar rendimiento.
- **SMT (Simultaneous Multithreading)**: varios hilos por núcleo (ej. Hyper-Threading de Intel).
- **Multicore**: varios núcleos físicos en un chip.

7. TENDENCIAS FUTURAS

- **Computación cuántica**: uso de **qubits**, permite paralelismo masivo y algoritmos no clásicos.
- **Arquitecturas neuromórficas**: diseñadas para imitar el cerebro humano.
- **Aceleradores de IA**:
    - **TPU (Tensor Processing Unit)**: de Google, optimizadas para redes neuronales.
    - **NPU (Neural Processing Unit)**: en dispositivos móviles.
    - **FPGAs (Field-Programmable Gate Arrays)**: configurables post-fabricación.
- **Sistemas heterogéneos**: combinación de CPU, GPU y otros aceleradores especializados.

APLICACIÓN DIDÁCTICA (CFGS DAM – Módulo "Programación de procesos y servicios")

1. REQUISITOS PREVIOS

2. OBJETIVOS DE APRENDIZAJE

- Analizar el impacto de la arquitectura del sistema en la ejecución de servicios.
- Programar de forma eficiente teniendo en cuenta núcleos, concurrencia y jerarquía de memoria.

3. METODOLOGÍA

- ABR (Aprendizaje Basado en Retos). Simulación de entornos reales.
- Uso de herramientas de análisis: htop, perf, taskset, systemd.

4. ATENCIÓN A LA DIVERSIDAD (Niveles III y IV).

5. DISEÑO UNIVERSAL PARA EL APRENDIZAJE (DUA)

- **Representación:** diagramas de arquitectura, vídeos explicativos.
- **Expresión:** scripts, paneles, presentaciones.
- **Compromiso:** retos prácticos contextualizados.

6. ACTIVIDAD PRINCIPAL
"Optimizando procesos según la arquitectura del sistema" *Proyecto práctico por equipos con Python.*
Fases:

1. **Análisis del sistema**: detección de arquitectura (núcleos, RAM) con os, platform, psutil.
2. **Programación concurrente**:
    - Con multiprocessing y threading.
    - Afinidad a núcleos con os.sched_setaffinity().
3. **Medición de rendimiento**:
    - Scripts instrumentados con time, tracemalloc.
    - Análisis con perf, htop, comparativa de configuraciones.
4. **Automatización**:
    - Scripts como demonios con systemd.
    - Monitorización de CPU, registro en log.
5. **Defensa y entrega**: Informe técnico + exposición oral con resultados y gráficos.