



CICLO: ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS

(Grado Superior)

MÓDULO: FUNDAMENTOS DE PROGRAMACIÓN

ÍNDICE

- 1. Capacidades Terminales**
- 2. Criterios de evaluación**
- 3. Contenidos**
- 4. Bibliografía**
- 5. Otras consideraciones**

1. Capacidades terminales

Al finalizar este módulo profesional, el alumno o la alumna deberá ser capaz de:

1. Elegir y definir una estructura de datos para resolver un problema con lenguajes estructurados.
2. Aplicar la metodología de desarrollo estructurado para el diseño de algoritmos.
3. Codificar programas en lenguajes estructurados de tercera generación.

2. Criterios de evaluación

1.- Al elegir y definir una estructura de datos para resolver un problema con lenguajes estructurados, el alumno o la alumna deberá ser capaz de:

- Describir las estructuras de datos típicas que maneja un lenguaje estructurado, su utilidad y ámbito de aplicación.
- Citar operaciones que permite realizar una estructura de datos desde un programa y explicar sus algoritmos.
- Justificar la importancia de la adecuada selección de estructuras de datos para la resolución de problemas en programación.
- Sobre un problema de programación en gestión propuesto:
 - Elegir las estructuras más adecuadas para representar y manejar los datos del problema.
 - Describir los algoritmos de tratamiento de las estructuras para la resolución del problema.
- Mostrar autonomía y decisión en la resolución de problemas con lenguajes estructurados.

2.- Al aplicar la metodología de desarrollo estructurado para el diseño de algoritmos, el alumno o la alumna deberá ser capaz de:

- Clasificar los lenguajes de programación según su nivel de abstracción y los recursos y procedimientos de desarrollo utilizados.
- Describir las características propias de la programación estructurada y justificar las ventajas que comporta.
- Identificar las estructuras básicas de programación.
- Definir las condiciones, el modo de aplicación de algún método de programación estructurada y la sintaxis de un lenguaje gráfico de representación de algoritmos.
- Clasificar las instrucciones típicas de los lenguajes estructurados según su función.
- Sobre un problema de programación en gestión propuesto:
 - Identificar y definir las estructuras de datos propias del problema.
 - Elaborar y representar un algoritmo aplicando métodos de programación estructurada.
 - Elaborar un conjunto de datos de prueba del programa diseñado.
- Mostrar preocupación por el orden y el método en la codificación de programas.
- Valorar la necesidad de la formación permanente y la adaptación a los cambios tecnológicos.

3.- Al codificar programas en lenguajes estructurados de tercera generación, el alumno o la alumna deberá ser capaz de:

- Interpretar la sintaxis del lenguaje y sus instrucciones.
- Definir las instrucciones, funciones y librerías del lenguaje más básicas y su utilidad.
- Describir el entorno de desarrollo del lenguaje: recursos que se utilizan y procedimiento práctico de desarrollo de programas.
- En un supuesto en el que se dispone de un sistema y de la documentación de referencia del lenguaje y un programa ya diseñado que responde a un problema propuesto:
 - Interpretar correctamente la información que suministran los manuales.
 - Codificar un programa fuente en el lenguaje con comentarios significativos y concisos, que defina adecuadamente las estructuras de datos y utilice correctamente las instrucciones, funciones y librerías del lenguaje.
 - Depurar el programa fuente y obtener un programa ejecutable.
- Mostrar una actitud positiva hacia el trabajo en equipo.
- Respetar las normas y especificaciones establecidas en la codificación de programas.

3. Contenidos

Los contenidos que se presentan a continuación están organizados en bloques.

Bloque I : Estructuras de datos y metodología de programación.

Bloque II : Codificación de programas en lenguajes estructurados.

Bloque I: ESTRUCTURAS DE DATOS Y METODOLOGÍA DE PROGRAMACIÓN

Este bloque abarca los contenidos derivados de las capacidades 1 y 2 que tratan de capacitar al alumno o alumna en la utilización de metodologías adecuadas de programación.

Este primer bloque de contenidos es básico para el correcto desarrollo del segundo bloque.

Procedimentales:

- Definición y uso de variables elementales.
- Definición y uso de variables estructuradas: registros, ficheros, arrays, listas, árboles.
- Uso de instrucciones básicas.
- Interpretación de algoritmos.
- Utilización de una metodología de desarrollo estructurado para el diseño de algoritmos.
- Análisis de las estructuras de datos utilizadas.
- Análisis de los algoritmos utilizados.
- Elaboración de la documentación de la programación.
- Creación y utilización de subprogramas.
- Definición y uso de argumentos en los subprogramas.

Hechos, conceptos y principios:

- Experiencia profesional y formación continua: trayectorias de promoción profesional, reciclaje más habitual, instituciones que lo imparten, estudios universitarios y no universitarios asociados.
- Ocupaciones relacionadas con las competencias profesionales: puestos de trabajo, condiciones de trabajo, requisitos de acceso más característicos.
- Concepto de programa.

- Tipos de programas y de lenguajes: interactivos y por lotes (batch), compilados e interpretados, alto o bajo nivel, etc.
- Especificaciones del programa o aplicación a desarrollar.
- Metodología de la programación:
 - Algoritmos.
 - Programación estructurada.
 - Programación modular.
 - Recursividad.
- Estructuras de datos:
 - Estructuras estáticas de datos.
 - Estructuras dinámicas de datos.
- Importancia de la documentación de los programas.

Actitudinales :

- Autonomía y decisión dentro del marco de las funciones y objetivos asignados por sus superiores.
- Orden y método a la hora de abordar su trabajo.

Bloque II: CODIFICACIÓN DE PROGRAMAS EN LENGUAJES ESTRUCTURADOS

Este bloque abarca contenidos derivados principalmente de la capacidad terminal 3, que trata de capacitar al alumno o alumna en la codificación de programas utilizando para ello lenguajes estructurados de tercera generación. En la adecuada consecución de estas capacidades se ven implicados algunos contenidos correspondientes a las capacidades terminales 1 y 2 que aparecen en el bloque anterior. Este es un bloque que necesita de el bloque anterior para poder desarrollarse.

Procedimentales:

- Uso del entorno de programación utilizado.
- Uso de la ayuda.
- Uso de otras utilidades de desarrollo: editores, enlazadores, traductores, depuradores, librerías normalizadas, etc.
- Utilización de manuales de referencia y otras fuentes de información.
- Codificación de programas en el lenguaje utilizado:
 - definición y uso de variables.
 - definición y uso de estructuras de datos estáticas.
 - definición y uso de estructuras de datos dinámicas.
 - definición y uso de subprogramas.
 - uso de parámetros.
- Utilización de una metodología de desarrollo estructurado para el diseño de algoritmos.
- Implementación de las estructuras de datos y de los algoritmos elegidos en el lenguaje utilizado.
- Creación y mantenimiento de librerías de subprogramas.
- Depuración de los módulos creados.
- Integración de los distintos módulos que forman parte del programa.
- Realización de pruebas de funcionamiento de cada módulo individualmente.
- Realización de pruebas de la integración entre los módulos.
- Comprobación de los formatos de entrada y salida del programa.
- Validación del cumplimiento de las especificaciones del programa.
- Evaluación de la eficiencia del programa.
- Medición del consumo de recursos del programa.
- Elaboración de los comentarios para el código fuente desarrollado.
- Análisis de las estructuras de datos utilizadas.
- Análisis de los algoritmos utilizados.

- Elaboración de matrices de referencias cruzadas entre procesos y almacenamiento de datos.
- Redacción de guías de uso de los programas y aplicaciones.
- Valoración del lenguaje empleado.

Hechos, conceptos y principios:

- Especificaciones de los programas y aplicaciones a desarrollar.
- Características de las estructuras de datos disponibles en el lenguaje empleado.
- Características de las instrucciones de control disponibles en el lenguaje empleado.
- Tipos y características de los subprogramas en el lenguaje empleado.
- Ventajas de la reutilización del código y el uso de librerías de subprogramas.
- Calidad de los programas y aplicaciones.
- Necesidad y conveniencia de la prueba y validación de los programas desarrollados.

Actitudinales :

- Valoración de la necesidad de autoestudio y formación complementaria durante su vida laboral.
- Actitud abierta ante los cambios que la evolución tecnológica ocasionan en el desarrollo profesional.
- Disposición favorable para el trabajo en equipo y la coordinación con otros equipos de trabajo.
- Respeto por las normas, convenciones y especificaciones impuestas a su trabajo de programación.
- Respeto por la integridad y confidencialidad de los datos manejados.
- Interés por la claridad y legibilidad de los programas.
- Compromiso con la calidad en el desarrollo de códigos.

4. Bibliografía

- ❑ "C/C++" Francisco Javier Ceballos Editorial RAMA
- ❑ "Programación Estructurada y Fundamentos de Programación" Sanchez-Chamoro-Molina-Matellan Editorial Mc Graw-Hill
- ❑ "Programación en lenguajes estructurados" Quero Catalinas-Lopez Herranz Paraninfo
- ❑ "Borlan C++ 4/4.5 Iniciación y Referencia" Joyanes Mc Graw-Hill

5. Otras consideraciones

- La duración del módulo en horas al año es de 285.
- El examen constará de ejercicios prácticos (pero no en ordenador), en el que se solicitarán pseudocódigo u ordinogramas y la codificación correspondiente en C++.
- La Versión que se utilizará para el Borlan C++ es la 5.