



CICLO: DESARROLLO DE APLICACIONES INFORMÁTICAS

(Grado Superior)

***MÓDULO: PROGRAMACIÓN EN LENGUALES
ESTRUCTURADOS***

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. Elaborar programas utilizando lenguajes estructurados, cumpliendo con las especificaciones establecidas en el diseño.
2. Evaluar el funcionamiento de las aplicaciones mediante la realización de pruebas de los diferentes módulos de programación.
3. Elaborar la documentación completa relativa a las aplicaciones desarrolladas facilitando su posterior mantenimiento, actualización y reutilización.
4. Adaptar aplicaciones a partir de nuevos requerimientos establecidos en el diseño.

2. Criterios de evaluación

1. Al elaborar programas utilizando lenguajes estructurados, cumpliendo con las especificaciones establecidas en el diseño, el alumno o la alumna deberá ser capaz de:

- Respetar las especificaciones iniciales del diseño.
- Trabajar en equipo en el desarrollo de los distintos módulos que forman parte de un mismo programa.
- Obtener información por sus propios medios de los manuales y de la ayuda del entorno de programación utilizado.
- Diferenciar la programación de procesos interactivos y por lotes (batch).
- Describir la utilidad de las librerías y de los enlazadores de los sistemas operativos y depuradores, así como su forma de empleo.
- Evaluar la importancia de la claridad y legibilidad de los programas para facilitar el mantenimiento y el trabajo en equipo.
- Elegir y definir las estructuras de datos necesarias para la resolución del problema en un lenguaje estructurado.
- Aplicar una metodología de desarrollo estructurado para el diseño de algoritmos.
- Codificar programas en un lenguaje estructurado a partir de los algoritmos diseñados.
- Aplicar estrategias de programación modular y de programación orientada a objetos, utilizando módulos, unidades o paquetes de desarrollo de programas.
- Reutilizar el código creado anteriormente en forma de módulos, unidades y librerías de una manera eficiente.
- Codificar un módulo de programación en un lenguaje estructurado de tercera generación.
- Documentar el código de un módulo de programación con comentarios significativos, concisos y legibles.
- Integrar y enlazar módulos de programación, rutinas y utilidades, siguiendo las especificaciones del diseño y el diagrama de estructuras.

- Comprobar que la utilización de recursos del sistema (procesador, memoria, periféricos) permiten que la integración y el enlace de programas sea ejecutable.

- Depurar los módulos de programación siguiendo criterios de eficiencia y eficacia.

2. Al evaluar el funcionamiento de las aplicaciones mediante la realización de pruebas de los diferentes módulos de programación, el alumno o la alumna deberá ser capaz de:

- Describir tipologías de errores de los tratamientos de una manera ordenada y metódica.

- Aplicar estándares de control de calidad a partir de las especificaciones establecidas en el diseño y de las prestaciones esperadas por el usuario de la aplicación.

- Realizar pruebas para cada módulo de una aplicación y pruebas de integración.

- Comprobar que el acceso y el tratamiento de los datos cumplen las directrices de la administración de datos del sistema y las especificaciones establecidas en el diseño.

- Comprobar que los formatos de entrada y salida de la aplicación son los esperados.

- Comprobar que la funcionalidad de los datos es la esperada.

- Provocar y verificar los diversos tratamientos de error.

- Validar que el funcionamiento de las aplicaciones desarrolladas es conforme a las especificaciones y requisitos de partida.

- Medir los rendimientos de la aplicación y evaluar la eficiencia de las prestaciones de la aplicación y el consumo de recursos.

3. Al elaborar la documentación completa relativa a las aplicaciones desarrolladas facilitando su posterior mantenimiento, actualización y reutilización, el alumno o la alumna deberá ser capaz de:

- Comentar el código creado de una manera breve y concisa, pero legible y clarificadora.

- Documentar y describir las estructuras de datos utilizadas.

- Elaborar matrices de referencias cruzadas que relacionen procesos y almacenamientos de datos.

- Redactar guías de uso de las aplicaciones.

- Facilitar la posterior adaptación o reutilización de las aplicaciones desarrolladas.

4. Al adaptar aplicaciones a partir de nuevos requerimientos establecidos en el diseño, el alumno o la alumna deberá ser capaz de:

- Respetar el trabajo realizado por el creador de la aplicación a adaptar.

- Utilizar las nuevas herramientas y metodologías surgidas desde la creación inicial de la aplicación a modificar mejorándola así.

- Identificar los datos y módulos de programación afectados por la modificación de los requerimientos.

- Utilizar matrices de referencias cruzadas entre módulos y datos para conocer las implicaciones que sobre éstos producen las nuevas modificaciones.

- Probar que los nuevos datos y módulos no producen pérdidas de eficiencia y eficacia en las prestaciones ni en los rendimientos de la aplicación y satisfacen los nuevos requerimientos funcionales.

- Documentar los cambios realizados sobre los datos, módulos y estructuras de datos y control de la aplicación.
- Mantener librerías de módulos reutilizables.
- Estudiar la documentación disponible para implementar correctamente las modificaciones deseadas.
- Codificar las modificaciones atendiendo a las interacciones con el resto del código ya existente.

3. Contenidos

Los contenidos que se presentan a continuación están organizados en bloques. Se ha optado por hacer unos bloques de contenidos que reflejan nítidamente las dos funciones más relevantes que forman parte del ejercicio profesional característico de este módulo:

- Bloque I : Creación y desarrollo de nuevos programas
- Bloque II : Mantenimiento y adaptación de software existente

Bloque I: CREACIÓN Y DESARROLLO DE NUEVOS PROGRAMAS

Este bloque abarca los contenidos derivados de las capacidades 1, 2 y 3, que tratan de capacitar al alumno o alumna en la creación y desarrollo completo de nuevos programas a partir de las especificaciones establecidas, ocupándose tanto de la creación inicial del programa, como de la evaluación de su funcionamiento y de la documentación del programa.

Este primer bloque de contenidos es claramente más importante y extenso que el segundo bloque, da respuesta a la principal actividad profesional de este módulo.

Procedimentales:

- Elaboración de programas:
 - 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: estructuras de control, funciones y subprogramas, etc.
 - Utilización de una metodología de desarrollo estructurado para el diseño de algoritmos.
 - Definición de las estructuras de datos necesarias en un lenguaje estructurado.
 - 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.
 - Creación de programas con lenguajes orientados a objetos.
- Evaluación de programas:
 - Pruebas de cada módulo individualmente.
 - 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.
- Documentación de programas:

- Comentarios adecuados sobre el código fuente del programa.
- Descripción de las estructuras de datos utilizadas.
- Descripción 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.

Hechos, conceptos y principios:

- Ocupaciones relacionadas con las competencias profesionales: puestos de trabajo, condiciones de trabajo, requisitos de acceso más característicos.
- Experiencia profesional y formación continua: trayectorias de promoción profesional, reciclaje más habitual, instituciones que lo imparten, estudios universitarios y no universitarios asociados.
- 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.
 - Programación orientada al objeto.
- Estructuras de datos:
 - Estructuras estáticas de datos.
 - Estructuras dinámicas de datos.

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.
- Capacidad de autonomía y decisión en el entorno laboral y dentro del marco de las funciones y objetivos asignados por sus superiores.
- Orden y método a la hora de abordar su trabajo.
- Actitud positiva 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.
- Interés por la claridad y legibilidad de los programas.

Bloque II: MANTENIMIENTO Y ADAPTACIÓN DE SOFTWARE EXISTENTE

Este bloque abarca contenidos derivados principalmente de la capacidad terminal 4, que trata de capacitar al alumnado en el mantenimiento y adaptación de programas anteriores. También se ven implicados en este bloque algunos contenidos de las capacidades 2 y 3 que aparecen en el bloque anterior. Este es un bloque con muchos menos contenidos que el anterior, pero que necesita de él para poder desarrollarse.

Procedimentales:

- Análisis de la documentación y de las matrices de referencias cruzadas entre módulos y datos del programa a modificar.
- Análisis de los nuevos requerimientos funcionales que se quieren implementar.
- Identificación de lo que hay que modificar o añadir al programa original.
- Codificación de las modificaciones atendiendo a las posibles interacciones con el programa original.
- Prueba y evaluación del funcionamiento de las modificaciones.
- Evaluación del funcionamiento del programa final atendiendo a los posibles efectos indeseados producidos por las modificaciones.
- Documentación conveniente de todas las modificaciones realizadas.
- Modificación de la documentación sobre los aspectos del programa original que hayan sido afectados por las modificaciones.
- Mantenimiento de librerías de subprogramas.

Hechos, conceptos y principios:

- Especificaciones del programa o aplicación a modificar o actualizar.
- Estructuras de datos.
- Algoritmos.
- Metodología de programación en el mantenimiento y modificación del software.

Actitudinales:

- Respeto por el trabajo a modificar y por los derechos de propiedad intelectual.
- Actitud cuidadosa ante los efectos de su trabajo.

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
- ❑ "Visual C++ 6 Curso de Iniciación" Edgar D'Andrea Infor BOOKS

5. Otras consideraciones

- La duración del módulo en horas al año es de 355.
- El exámen 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.