

ANEXO X: MÓDULOS OPTATIVOS

ANEXO X-XVII. FAMILIA PROFESIONAL INFORMÁTICA Y COMUNICACIONES

MÓDULO: INTRODUCCIÓN A LA PROGRAMACIÓN

DURACIÓN: 96 HORAS

CFGM SISTEMAS MICROINFORMÁTICOS Y REDES

CFGS ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED

RESULTADOS DE APRENDIZAJE Y CRITERIOS DE EVALUACIÓN

RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.

Se han identificado los bloques que componen la estructura de un programa informático. Se han creado proyectos de desarrollo de aplicaciones

Se han utilizado entornos integrados de desarrollo.

Se han identificado los distintos tipos de variables y la utilidad específica de cada uno. Se ha modificado el código de un programa para crear y utilizar variables.

Se han creado y utilizado constantes y literales.

Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje. Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas. Se han introducido comentarios en el código.

RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.

Se han identificado los fundamentos de la programación orientada a objetos. Se han escrito programas simples.

Se han instanciado objetos a partir de clases predefinidas. Se han utilizado métodos y propiedades de los objetos.

Se han escrito llamadas a métodos estáticos.

Se han utilizado parámetros en la llamada a métodos. Se han incorporado y utilizado librerías de objetos.

Se han utilizado constructores.

Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.

RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.

Se ha escrito y probado código que haga uso de estructuras de selección.

Se han utilizado estructuras de repetición.

Se han reconocido las posibilidades de las sentencias de salto. Se ha escrito código utilizando control de excepciones.

Se han creado programas ejecutables utilizando diferentes estructuras de control. Se han probado y depurado los programas.

Se ha comentado y documentado el código. Se han creado excepciones.

Se han utilizado aserciones para la detección y corrección de errores durante la fase de desarrollo.

RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.

Se ha reconocido la sintaxis, estructura y componentes típicos de una clase. Se han definido clases.

Se han definido propiedades y métodos. Se han creado constructores.

Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.

Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros. Se han definido y utilizado clases heredadas.

Se han creado y utilizado métodos estáticos.

Se han creado y utilizado conjuntos y librerías de clases.

RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

Se ha utilizado la consola para realizar operaciones de entrada y salida de información.

Se han aplicado formatos en la visualización de la información.

Se han reconocido las posibilidades de entrada / salida del lenguaje y las librerías asociadas.

Se han utilizado ficheros para almacenar y recuperar información.

Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.

Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples.

Se han programado controladores de eventos.

Se han escrito programas que utilicen interfaces gráficos para la entrada y salida de información.

RA6. Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos.

Se han escrito programas que utilicen matrices (arrays).

Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados. Se han utilizado listas para almacenar y procesar información.

Se han utilizado iteradores para recorrer los elementos de las listas.

Se han reconocido las características y ventajas de cada una de las colecciones de datos disponibles.

Se han creado clases y métodos genéricos.

Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.

Se han identificado las clases relacionadas con el tratamiento de documentos escritos en diferentes lenguajes de intercambio de datos.

Se han realizado programas que realicen manipulaciones sobre documentos escritos en diferentes lenguajes de intercambio de datos.

Se han utilizado operaciones agregadas para el manejo de información almacenada en colecciones.

CONTENIDOS ORIENTATIVOS:

Identificación de los elementos de un programa informático:

Estructura y bloques fundamentales. Variables. Tipos de datos. Literales. Constantes. Operadores y expresiones. Conversiones de tipo. Comentarios.

Utilización de objetos:

Características de los objetos. Instanciación de objetos. Utilización de métodos. Parámetros. Utilización de propiedades. Utilización de métodos estáticos.

Constructores. Destrucción de objetos y liberación de memoria.

Uso de estructuras de control:

Estructuras de selección. Estructuras de repetición. Estructuras de salto. Control de excepciones. Aserciones. Prueba, depuración y documentación de la aplicación.

Desarrollo de clases:

Concepto de clase. Estructura y miembros de una clase. Visibilidad. Creación de propiedades. Creación de métodos. Creación de constructores. Utilización de clases y objetos. Utilización de clases heredadas.

Lectura y escritura de información:

Flujos. Tipos: bytes y caracteres. Clases relacionadas. Ficheros de datos. Registros. Apertura y cierre de ficheros. Modos de acceso. Escritura y lectura de información en ficheros. Utilización de los sistemas de ficheros. Creación y eliminación de ficheros y directorios. Entrada desde teclado. Salida a pantalla. Formatos de visualización. Interfaces gráficas. Concepto de evento. Creación de controladores de eventos.

Aplicación de las estructuras de almacenamiento:

Estructuras estáticas y dinámicas. Creación de matrices (arrays). Matrices (arrays) multidimensionales. Genericidad. Cadenas de caracteres. Expresiones regulares. Colecciones: Listas, Conjuntos y Diccionarios. Operaciones agregadas: filtrado, reducción y recolección.