

# PROGRAMACIÓN ANUAL

|                                  |                        |   |
|----------------------------------|------------------------|---|
| I.E.S. INGENIERO DE LA<br>CIERVA | PROGRAMACIÓN DIDÁCTICA | Departamento de la<br>Familia Profesional<br>de Informática |
|----------------------------------|------------------------|---|

MÓDULO: DESARROLLO DE INTERFACES  
Código: 0488

M<sup>a</sup> DEL MAR MARTÍN

CURSO 2019/2020

I.E.S. INGENIERO DE LA CIERVA - MURCIA

## INDICE

1. JUSTIFICACIÓN, 4
  - 1.1. LA PROGRAMACIÓN DIDACTICA. ¿POR QUÉ PROGRAMAR?, 5
2. CONTEXTUALIZACIÓN, 5
3. OBJETIVOS, 6
  - 3.1. OBJETIVOS GENERALES DEL CICLO, 6
  - 3.2. OBJETIVOS DEL MODULO/CAPACIDADES TERMINALES, 7
  - 3.3. CRITERIOS DE EVALUACIÓN, 8
  - 3.4. CORRELACIÓN DE LOS OBJETIVOS Y CRITERIOS DE EVALUACIÓN, 8
4. CONTENIDOS, 10
  - 4.1. JUSTIFICACIÓN DE LA SECUENCIACIÓN DE LAS UNIDADES DE TRABAJO
  - 4.2. SECUENCIACIÓN DE LAS UNIDADES DE TRABAJO, 11
  - 4.3. TEMPORALIZACIÓN DE LAS UNIDADES DE TRABAJO, 28
5. METODOLOGÍA, 29
  - 5.1. ORIENTACIONES METODOLÓGICAS, 29
  - 5.2. ESTRATEGIAS DE ENSEÑANZA-APRENDIZAJE, 29
  - 5.3. TIPOS DE ACTIVIDADES, 30
  - 5.4. ORGANIZACIÓN DEL AULA, ESPACIOS Y TIEMPOS, 30
  - 5.5. MATERIALES CURRICULARES Y RECURSOS DIDÁCTICOS, 30
6. CRITERIOS SOBRE EVALUACIÓN, PROCEDIMIENTOS E INSTRUMENTOS DE EVALUACIÓN, 30
  - 6.1. CRITERIOS DE EVALUACIÓN, 30
  - 6.2. PROCEDIMIENTOS E INSTRUMENTOS DE EVALUACIÓN DEL PROCESO DE APRENDIZAJE, 32
7. CRITERIOS DE CALIFICACIÓN, 32
  - 7.1. INDICADORES DE VALORACIÓN, 32
  - 7.2. CONVOCATORÍAS ORDINARIA Y EXTRAORDINARIA, 34
8. MEDIDAS DE ATENCIÓN A LA DIVERSIDAD, 342

9. TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES, 34
10. TEMAS TRANSVERSALES, 36
- 11 .PREVENCIÓN DE RIESGOS LABORALES, 36
12. ACTIVIDADES COMPLEMENTARIAS Y EXTRAESCOLARES, 36
13. EVALUACIÓN DE LOS PROCESOS DE LA ENSEÑANZA Y DE LA PRÁCTICA DOCENTE, 36

## 1. JUSTIFICACIÓN.

Programar consiste en establecer documentalmente, para un módulo formativo concreto, el conjunto de actuaciones y actividades, proyectadas a lo largo de un tiempo determinado, que el profesor y el alumnado van a desarrollar para alcanzar los objetivos establecidos. Dichos objetivos deben estar referenciados y basados en la Competencia/es Profesional/es asociada al módulo a programar, tal y como especifica el Real Decreto de desarrollo del título formativo.

Así pues, la programación de un módulo puede considerarse, como la planificación estructurada, ordenada y escrita del desarrollo del módulo. El resultado de la programación es un conjunto de unidades ordenadas y secuenciadas.

Teniendo en cuenta: la realidad del entorno socio-productivo; los espacios y medios disponibles; y las características concretas del alumnado, se definen múltiples aspectos, tales como:

- Objetivos: son las capacidades que el alumno debe alcanzar.
- Contenidos más adecuados a desarrollar para alcanzar los objetivos.
- Metodología: la forma más coherente de llevar a cabo la práctica educativa.
- Determinación de las unidades de trabajo, secuenciadas y temporalizadas, en que se va a dividir el módulo.
- Criterios de evaluación.
- Atención a la diversidad.
- Recursos didácticos y materiales necesarios.

entre otros.

En el Real Decreto de desarrollo del título formativo se expresan los objetivos de cada módulo, formulados en términos de Resultados de Aprendizaje (RA) que deben ser alcanzados por el alumnado y los Contenidos que deben abordarse para su logro. Con estos elementos debemos realizar la planificación, ordenación y secuenciación de los mismos, para que junto con las actividades de formación y evaluación, obtengamos como producto la programación de la actividad docente organizada y ordenada en unidades de trabajo.

Dicha programación y ordenación de los contenidos no es sino el establecimiento de lo que

podemos llamar un "Recorrido Didáctico", que es una forma de expresar y recoger cuáles y en qué momentos se van incorporando los contenidos establecidos en el módulo profesional para generar un proceso de enseñanza-aprendizaje continuado y coherente que garantice alcanzar los RA propuestos.

En aquellos módulos asociados a una unidad de competencia se establece una correspondencia biunívoca entre la unidad de competencia y su módulo. Por ello, a la hora de programar un módulo de estas características se analiza el perfil profesional. Este análisis consiste en hacer una lectura comprensiva de los elementos que conforman la unidad de competencia, ya que esto ayuda a referenciar el contexto laboral en el que algún día el alumno o la alumna deberá poner en valor su aprendizaje.

### 1.1. LA PROGRAMACIÓN DIDÁCTICA. ¿POR QUÉ PROGRAMAR?

La programación es una parte o aspecto del diseño del currículo cuya responsabilidad recae de modo directo en el profesorado de un departamento y que es consecuencia y viene limitada por una cadena de decisiones curriculares previas adoptadas en otros ámbitos o subsistemas de decisión curricular.

Desde la perspectiva de la planificación previa, la función esencial que la programación o el diseño del currículo pretenden desempeñar es, básicamente, la de proponer un plan de acción que guíe y oriente los procesos y las actividades de enseñanza-aprendizaje al objeto de conseguir los objetivos propuestos.

¿Para qué sirve programar? Como instrumento de planificación, reflexión y orientación, evidentemente la programación sirve en definitiva como elemento que permite aumentar la conciencia del profesor sobre su práctica educativa y por tanto como instrumento a través del cual se articula la calidad de la educación.

## 2. CONTEXTUALIZACIÓN.

Desarrollado en los puntos **1.2, 1.4-1.8** de la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.

### 3. OBJETIVOS

La Formación Profesional tiene como objetivos conseguir que los alumnos y las alumnas adquieran las capacidades que les permitan:

- a) Desarrollar la competencia general correspondiente a la cualificación o cualificaciones objeto de los estudios realizados.
- b) Comprender la organización y características del sector productivo correspondiente, así como los mecanismos de inserción profesional; conocer la legislación laboral y los derechos y obligaciones que se derivan de las relaciones laborales.
- c) Aprender por sí mismos y trabajar en equipo, así como formarse en la prevención de conflictos y en la resolución pacífica de los mismos en todos los ámbitos de la vida personal, familiar y social.
- d) Trabajar en condiciones de seguridad y salud, así como prevenir los posibles riesgos derivados del trabajo.
- e) Desarrollar una identidad profesional motivadora de futuros aprendizajes y adaptaciones a la evolución de los procesos productivos y al cambio social.
- f) Afianzar el espíritu emprendedor para el desempeño de actividades e iniciativas profesionales.
- g) Lograr las competencias relacionadas con las áreas prioritarias referidas en la Ley Orgánica 5/2002, de 19 de junio, de las Cualificaciones y de la Formación Profesional.
- h) Hacer realidad la formación a lo largo de la vida y utilizar las oportunidades de aprendizaje a través de las distintas vías formativas para mantenerse actualizado en los distintos ámbitos: social, personal, cultural y laboral, conforme a sus expectativas, necesidades e intereses.
- i) Fomentar la igualdad de oportunidades entre hombres y mujeres para acceder a una formación que permita todo tipo de opciones profesionales y el ejercicio de las mismas.

Este módulo contribuye al logro de los objetivos citados anteriormente, como se detalla a continuación.

#### 3.1. OBJETIVOS GENERALES DEL CICLO

***UBICACIÓN DEL CICLO "DESARROLLO DE APLICACIONES MULTIPLATAFORMA" EN EL SISTEMA EDUCATIVO LOE/LOMCE***

Desarrollado en el punto **1.3** de la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.

### ***Competencia general***

La competencia general de este título consiste en desarrollar, implantar, documentar y mantener aplicaciones informáticas multiplataforma, utilizando tecnologías y entornos de desarrollo específicos, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de «usabilidad» y calidad exigidas en los estándares establecidos.

Este módulo corresponde a la cualificación y unidad de competencia del Catálogo Nacional de Cualificaciones Profesionales siguiente:

- Desarrollo de Interfaces (R.D. 1087/2005, de 16 de septiembre), que comprende las siguientes unidades de competencia:
  - UC0494\_3. Desarrollar componentes software en lenguajes de programación estructurada.

### ***COMPETENCIAS PROFESIONALES, PERSONALES Y SOCIALES***

Desarrollado en la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.

## **3.2. OBJETIVOS DEL MODULO/CAPACIDADES TERMINALES**

La formación profesional actual es una formación basada en competencias. Sintetizando, esto quiere decir que el currículo se establece en orden a alcanzar unas determinadas competencias que son requeridas para capacitar al alumnado en el desempeño de una actividad profesional. Esta competencia se describe en grupos o conjuntos de realizaciones profesionales, que expresan los resultados que se esperan del desempeño profesional. A cada uno de estos grupos se les denomina Unidad de Competencia. Los elementos que constituyen un módulo profesional son, pues, diseñados para alcanzar unos resultados en el proceso de aprendizaje, que se han denominado RA, que garanticen la adquisición de la competencia que se establece y describe en la Unidad de Competencia correspondiente.

Por ello, en el real decreto de este título se detallan para este módulo los objetivos en términos de Resultados de Aprendizaje (RA) como sigue:

- RA1. Genera interfaces gráficos de usuario mediante editores visuales utilizando las funcionalidades del editor y adaptando el código generado.
- RA2. Genera interfaces gráficos de usuario basados en XML utilizando herramientas específicas y adaptando el documento XML generado.
- RA3. Crea componentes visuales valorando y empleando herramientas específicas.
- RA4. Diseña interfaces gráficos identificando y aplicando criterios de usabilidad.
- RA5. Crea informes evaluando y utilizando herramientas gráficas.
- RA6. Documenta aplicaciones seleccionando y utilizando herramientas específicas.
- RA7. Prepara aplicaciones para su distribución evaluando y utilizando herramientas específicas.
- RA8. Evalúa el funcionamiento de aplicaciones diseñando y ejecutando pruebas.

### 3.3. CRITERIOS DE EVALUACIÓN

Los criterios de evaluación se enuncian a continuación correlacionados con los objetivos, formulados como resultados de aprendizaje, pues son numerosos y se entienden mejor agrupados por el RA que jerárquicamente les sirve de referencia.

### 3.4. CORRELACIÓN DE LOS OBJETIVOS Y CRITERIOS DE EVALUACIÓN

#### **RA1. Genera interfaces gráficos de usuario mediante editores visuales utilizando las funcionalidades del editor y adaptando el código generado.**

##### **Criterios de evaluación:**

- a) Se ha creado un interfaz gráfico utilizando los asistentes de un editor visual.
- b) Se han utilizado las funciones del editor para ubicar los componentes del interfaz.
- c) Se han modificado las propiedades de los componentes para adecuarlas a las necesidades de la aplicación.
- d) Se ha analizado el código generado por el editor visual.
- e) Se ha modificado el código generado por el editor visual.
- f) Se han asociado a los eventos las acciones correspondientes.
- g) Se ha desarrollado una aplicación que incluye el interfaz gráfico obtenido.

#### **RA2. Genera interfaces gráficos de usuario basados en XML utilizando herramientas específicas y adaptando el documento XML generado.**

##### **Criterios de evaluación:**

- a) Se han reconocido las ventajas de generar interfaces de usuario a partir de su descripción en XML.
- b) Se ha generado la descripción del interfaz en XML usando un editor gráfico.



- c) Se ha analizado el documento XML generado.
- d) Se ha modificado el documento XML.
- e) Se han asignado acciones a los eventos.
- f) Se ha generado el código correspondiente al interfaz a partir del documento XML.
- g) Se ha programado una aplicación que incluye el interfaz generado.

**RA3. Crea componentes visuales valorando y empleando herramientas específicas.**

**Criterios de evaluación:**

- a) Se han identificado las herramientas para diseño y prueba de componentes.
- b) Se han creado componentes visuales.
- c) Se han definido sus propiedades y asignado valores por defecto.
- d) Se han determinado los eventos a los que debe responder el componente y se les han asociado las acciones correspondientes.
- e) Se han realizado pruebas unitarias sobre los componentes desarrollados.
- f) Se han documentado los componentes creados.
- g) Se han empaquetado componentes.
- h) Se han programado aplicaciones cuyo interfaz gráfico utiliza los componentes creados.

**RA4. Diseña interfaces gráficos identificando y aplicando criterios de usabilidad.**

**Criterios de evaluación:**

- a) Se han creado menús que se ajustan a los estándares.
- b) Se han creado menús contextuales cuya estructura y contenido siguen los estándares establecidos.
- c) Se han distribuido las acciones en menús, barras de herramientas, botones de comando, entre otros, siguiendo un criterio coherente.
- d) Se han distribuido adecuadamente los controles en la interfaz de usuario.
- e) Se ha utilizado el tipo de control más apropiado en cada caso
- f) Se ha diseñado el aspecto de la interfaz de usuario (colores y fuentes entre otros) atendiendo a su legibilidad.
- g) Se ha verificado que los mensajes generados por la aplicación son adecuados en extensión y claridad.
- h) Se han realizado pruebas para evaluar la usabilidad de la aplicación.

**RA5. Crea informes evaluando y utilizando herramientas gráficas.**

**Criterios de evaluación:**

- a) Se ha establecido la estructura del informe.
- b) Se han generado informes básicos a partir de una fuente de datos mediante asistentes.
- c) Se han establecido filtros sobre los valores a presentar en los informes.
- d) Se han incluido valores calculados, recuentos y totales.
- e) Se han incluido gráficos generados a partir de los datos.
- f) Se han utilizado herramientas para generar el código correspondiente a los informes de una aplicación.
- g) Se ha modificado el código correspondiente a los informes.

h) Se ha desarrollado una aplicación que incluye informes incrustados.

**RA6. Documenta aplicaciones seleccionando y utilizando herramientas específicas.**

**Criterios de evaluación:**

- a) Se han identificado sistemas de generación de ayudas.
- b) Se han generado ayudas en los formatos habituales.
- c) Se han generado ayudas sensibles al contexto.
- d) Se ha documentado la estructura de la información persistente.
- e) Se ha confeccionado el manual de usuario y la guía de referencia.
- f) Se han confeccionado los manuales de instalación, configuración y administración.
- g) Se han confeccionado tutoriales.

**RA7. Prepara aplicaciones para su distribución evaluando y utilizando herramientas específicas.**

**Criterios de evaluación:**

- a) Se han empaquetado los componentes que requiere la aplicación.
- b) Se ha personalizado el asistente de instalación
- c) Se ha empaquetado la aplicación para ser instalada de forma típica, completa o personalizada.
- d) Se han generado paquetes de instalación utilizando el entorno de desarrollo.
- e) Se han generado paquetes de instalación utilizando herramientas externas.
- f) Se han generado paquetes instalables en modo desatendido.
- g) Se ha preparado el paquete de instalación para que la aplicación pueda ser correctamente desinstalada.
- h) Se ha preparado la aplicación para ser descargada desde un servidor web y ejecutada.

**RA8. Evalúa el funcionamiento de aplicaciones diseñando y ejecutando pruebas.**

**Criterios de evaluación:**

- a) Se ha establecido una estrategia de pruebas.
- b) Se han realizado pruebas de integración de los distintos elementos.
- c) Se han realizado pruebas de regresión.
- d) Se han realizado pruebas de volumen y estrés.
- e) Se han realizado pruebas de seguridad.
- f) Se han realizado pruebas de uso de recursos por parte de la aplicación.
- g) Se ha documentado la estrategia de pruebas y los resultados obtenidos.

## 4. CONTENIDOS

Los contenidos son los elementos de referencia utilizados para alcanzar los resultados de aprendizaje del módulo. Expresan los procedimientos, conocimientos y actitudes que han de ser adquiridos a lo largo del proceso de aprendizaje. Van a servir de soporte y guía de las actividades

de enseñanza aprendizaje.

Esta tipología de contenidos se relaciona con los resultados de aprendizaje que promueven.

#### 4.1. JUSTIFICACIÓN DE LA SECUENCIACIÓN DE LAS UNIDADES DE TRABAJO

A la hora de determinar las unidades didácticas vamos a desarrollar un recorrido didáctico que tome como guía del mismo la descripción de la competencia a la que da respuesta el módulo. Establecer un recorrido didáctico significa, básicamente, determinar las Unidades Didácticas que conforman el módulo, su secuenciación y temporalización.

Al determinar las unidades didácticas se han tenido en cuenta los siguientes criterios:

- Cada unidad didáctica tiene una coherencia interna derivada de la necesidad de dar respuesta a una situación característica del desempeño referenciada en el perfil profesional.
- Las actividades propuestas van a servir para construir el aprendizaje de un modo significativo para el alumno, de forma que promuevan la adquisición de capacidades que permitan su transferencia a otros contextos y situaciones.
- Las actividades e instrumentos de evaluación van a dar respuesta al qué y cómo se evalúa, teniendo en cuenta los resultados de aprendizaje a alcanzar, los contenidos y los criterios de evaluación de cada módulo y el contexto en el que se desarrolla el proceso de enseñanza aprendizaje.

#### 4.2. SECUENCIACIÓN DE LAS UNIDADES DE TRABAJO

Secuenciar los contenidos es decidir en qué momento del proceso formativo va a ser trabajado cada uno de ellos para que sean aprendidos (significativamente) por los alumnos y por las alumnas y de este modo poder reestructurar y construir sus conocimientos.

La verdadera secuenciación se realiza cuando se decide de qué unidades didácticas va a constar el módulo y que objetivos se persiguen en cada una de ellas, de manera que los resultados de aprendizaje que se alcancen sean los propuestos. Y es lo que se presenta a continuación.

## U.T. 0. Presentación del módulo

### Objetivos de aprendizaje:

- Conocer la planificación global de desarrollo del módulo, así como a los miembros del grupo.
- Comprender los criterios que serán considerados y aplicados por el profesor o profesora en la gestión del proceso formativo.
- Identificar los derechos y obligaciones como estudiante, en relación con el módulo.
- Comprender las principales interrelaciones que se dan entre las unidades didácticas del módulo y entre este y los demás que lo constituyen.
- Identificar los propios conocimientos en relación con los que se deben alcanzar en el módulo.

### Contenidos

- Procedimentales
  - Análisis de las relaciones existentes entre los módulos del ciclo y las de éste con las cualificaciones que le sirven de referente.
  - Identificación y registro en el soporte adecuado de los aspectos, normas y elementos que se planteen en torno a cuestiones disciplinares, metodológicas, relacionales, etc.
- Conceptuales
  - Cualificaciones que constituyen el ciclo y relación con el módulo.
  - Contribución del módulo al logro de los objetivos del ciclo
  - Objetivos del módulo
  - Criterios de evaluación del módulo y de las unidades didácticas.
- Actitudinales
  - Valorar la importancia de lograr un consenso en relación con los comportamientos deseados por parte de todos los componentes del grupo, incluido el profesor o la profesora.
  - Normas y criterios a seguir en el desarrollo del módulo.

## U.T.1. Introducción a interfaz gráfica de usuario con VisualBasic.Net

**RA1. Genera interfaces gráficos de usuario mediante editores visuales utilizando las funcionalidades del editor y adaptando el código generado.**

### Objetivos de aprendizaje:

- Iniciar un proyecto Visual Basic .NET basado en la plantilla *Microsoft Windows® Application*.
- Utilizar la barra de herramientas estándar, las herramientas de programación y las ventanas de programación del entorno de desarrollo para trabajar con las aplicaciones.
- Crear una aplicación sencilla en Visual Basic .NET.

### Contenidos

- **Conceptos básicos de .NET**
  - ¿Qué es la plataforma .NET?
  - Cómo funciona el .NET Framework
  - ¿Qué es Visual Studio .NET?
  - Cómo utilizar la página de inicio
  - ¿Qué es una plantilla de aplicación?
  - Cómo utilizar el Explorador de soluciones
- **Explorar el entorno de desarrollo**
  - Menús y Barras de herramientas
  - El Cuadro de herramientas
  - Diseñador de Windows Forms
  - Editor de código
  - Ventana de propiedades
  - Otras ventanas de programación
  - Sistema de ayuda en línea
- **Crear un proyecto Visual Basic .NET**
  - El proceso de desarrollo
  - Cómo crear el interfaz de usuario
  - Cómo establecer las propiedades de los controles
  - Cómo añadir código a los controles
  - Cómo guardar, ejecutar y generar la aplicación

### **Actividades de enseñanza-aprendizaje**

- Evaluación inicial de conocimientos previos.
- Crear un proyecto Visual Basic .NET
- Trabajar en el entorno de desarrollo
- Guardar, ejecutar y generar la aplicación

### **Criterios de evaluación**

- Se ha creado un interfaz gráfico utilizando los asistentes de un editor visual.
- Se han utilizado las funciones del editor para ubicar los componentes del interfaz.
- Se han modificado las propiedades de los componentes para adecuarlas a las necesidades de la aplicación.
- Se ha desarrollado una aplicación que incluye el interfaz gráfico obtenido.

### **U.T.2. Elementos del lenguaje. Variables y estructuras de datos.**

**RA1. Genera interfaces gráficos de usuario mediante editores visuales utilizando las**

## funcionalidades del editor y adaptando el código generado.

### Objetivos de aprendizaje:

- Describir los diversos tipos de datos que puede utilizar para declarar variables.
- Nombrar, declarar, inicializar y utilizar variables y constantes.
- Declarar variables con diferentes niveles de ámbito.
- Crear estructuras de datos definidas por el usuario.
- Convertir valores de variables de un tipo de datos en otro.
- Almacenar datos en matrices.

### Contenidos

- Introducción a los tipos de datos
  - ¿Qué es el sistema de tipos comunes?
  - Tipos de datos
  - Cómo escoger un tipo de datos
- Uso de variables
  - ¿Qué son las variables?
  - Cómo nombrar las variables
  - Cómo declarar las variables
  - Cómo afecta **Option Explicit** a las variables
  - Cómo asignar valores a las variables
  - Cómo utilizar las variables
  - Variables vs. Constantes
- Ámbito de una variable
  - ¿Qué es el ámbito?
  - Cómo declarar variables locales
  - Cómo declarar variables estáticas
  - Cómo declarar variables de módulo
  - Cómo configurar los niveles de acceso para las variables
- Convertir tipos de datos
  - ¿Cuáles son las funciones de conversión?
  - Cómo convertir explícitamente tipos de datos
  - Cómo funciona la conversión implícita de datos
- Crear y utilizar estructuras
  - ¿Qué son las estructuras?
  - Cómo declarar estructuras
  - Cómo utilizar estructuras
- Almacenar datos en una matriz
  - ¿Qué es una matriz?
  - Cómo declarar una matriz unidimensional
  - Cómo utilizar matrices multidimensionales

- Cómo cambiar el tamaño de las matrices

### **Actividades de enseñanza-aprendizaje**

- Evaluación inicial de conocimientos previos.
- Escoger tipos de datos
- Encontrar los errores en la declaración de variables
- Configurar niveles de acceso para variables
- Realizar conversión de variables
- Crear y utilizar estructuras
- Aplicación con matrices

### **Criterios de evaluación**

- Se ha creado un interfaz gráfico utilizando los asistentes de un editor visual.
- Se han utilizado las funciones del editor para ubicar los componentes del interfaz.
- Se han modificado las propiedades de los componentes para adecuarlas a las necesidades de la aplicación.
- Se ha analizado el código generado por el editor visual.
- Se ha modificado el código generado por el editor visual.
- Se ha desarrollado una aplicación que incluye el interfaz gráfico obtenido.

### **U.T.3. Funciones. Subrutinas y procedimientos.**

**RA1. Genera interfaces gráficos de usuario mediante editores visuales utilizando las funcionalidades del editor y adaptando el código generado.**

**RA3. Crea componentes visuales valorando y empleando herramientas específicas.**

#### **Objetivos de aprendizaje:**

- Diferenciar entre un procedimiento Sub y un procedimiento Function.
- Crear e invocar procedimientos Sub y Function.
- Escribir procedimientos en módulos para permitir la reutilización del código.
- Pasar argumentos por valor y por referencia.
- Utilizar funciones predefinidas en el código de nuestra aplicación.

#### **Contenidos**

- Crear procedimientos
  - ¿Qué son los procedimientos?
  - Cómo crear procedimientos Sub
  - Cómo crear procedimientos Function

- Cómo declarar argumentos en procedimientos
- Pasar argumentos
- Cómo utilizar argumentos opcionales
- Escribir código reutilizable
- Reutilización de código
- Uso de procedimientos
  - Cómo utilizar procedimientos **Sub**
  - Cómo utilizar procedimientos **Function**
  - Cómo pasar matrices a procedimientos
  - Cómo crear un **Sub Main**
- Uso de funciones predefinidas
  - Cómo utilizar la función **InputBox**
  - Cómo utilizar las funciones Date y Time
  - Cómo utilizar las funciones String
  - Cómo utilizar las funciones Format
  - Cómo utilizar las funciones Financial

### Actividades de enseñanza-aprendizaje

- Evaluación inicial de conocimientos previos.
- Crear una Función en un Módulo
- Utilización del valor devuelto de una función
- Crear un Sub Main
- Uso de funciones predefinidas

### Criterios de evaluación

- Se ha creado un interfaz gráfico utilizando los asistentes de un editor visual.
- Se han utilizado las funciones del editor para ubicar los componentes del interfaz.
- Se han modificado las propiedades de los componentes para adecuarlas a las necesidades de la aplicación.
- Se ha analizado el código generado por el editor visual.
- Se han asociado a los eventos las acciones correspondientes.
- Se ha modificado el código generado por el editor visual.
- Se ha desarrollado una aplicación que incluye el interfaz gráfico obtenido.
- Se han identificado las herramientas para diseño y prueba de componentes.
- Se han creado componentes visuales.
- Se han definido sus propiedades y asignado valores por defecto.
- Se han determinado los eventos a los que debe responder el componente y se les han asociado las acciones correspondientes.



- Se han programado aplicaciones cuyo interfaz gráfico utiliza los componentes creados.

#### U.T.4. Bucles y estructuras de decisión.

**RA1. Genera interfaces gráficos de usuario mediante editores visuales utilizando las funcionalidades del editor y adaptando el código generado.**

**RA3. Crea componentes visuales valorando y empleando herramientas específicas.**

#### Objetivos de aprendizaje:

- Crear fórmulas y expresiones condicionales utilizando operadores aritméticos, de comparación y lógicos.
- Utilizar instrucciones If...Then para evaluar si una condición es verdadera o falsa y dirigir el flujo del programa en consecuencia.
- Utilizar instrucciones Select Case para probar diferentes valores de la misma expresión y ejecutar las instrucciones correspondientes.
- Utilizar instrucciones Do...Loop para ejecutar instrucciones hasta o mientras se satisface una condición específica.
- Utilizar instrucciones For...Next para ejecutar instrucciones un determinado número de veces.
- Escoger la estructura de decisión o bucle adecuado en función de los requerimientos de la aplicación.

#### Contenidos

- Uso de expresiones condicionales
  - ¿Qué son las expresiones condicionales?
  - Cómo utilizar operadores aritméticos
  - Cómo utilizar operadores de comparación
  - Cómo utilizar operadores lógicos
  - Cómo combinar operadores de comparación y lógicos
- Uso de estructuras de decisión
  - Cómo utilizar instrucciones **If...Then**
  - Cómo utilizar instrucciones **If...Then...Else**
  - Cómo utilizar instrucciones **If...Then...ElseIf**
  - Cómo utilizar instrucciones **Select Case**
  - Directrices para elegir una estructura de decisión
- Uso de estructuras de bucle condicionales
  - Cómo utilizar instrucciones **For...Next**
  - Cómo utilizar instrucciones **For Each...Next**
  - Cómo utilizar instrucciones **Do...Loop**
  - Cómo utilizar instrucciones **Exit**

## Actividades de enseñanza-aprendizaje

- Evaluación inicial de conocimientos previos.
- Evaluar expresiones condicionales
- Elegir una estructura de decisión adecuada
- Diseño de estructuras de bucle

## Criterios de evaluación

- Se ha creado un interfaz gráfico utilizando los asistentes de un editor visual.
- Se han utilizado las funciones del editor para ubicar los componentes del interfaz.
- Se han modificado las propiedades de los componentes para adecuarlas a las necesidades de la aplicación.
- Se ha analizado el código generado por el editor visual.
- Se han asociado a los eventos las acciones correspondientes.
- Se ha modificado el código generado por el editor visual.
- Se ha desarrollado una aplicación que incluye el interfaz gráfico obtenido.
- Se han identificado las herramientas para diseño y prueba de componentes.
- Se han creado componentes visuales.
- Se han definido sus propiedades y asignado valores por defecto.
- Se han determinado los eventos a los que debe responder el componente y se les han asociado las acciones correspondientes.
- Se han programado aplicaciones cuyo interfaz gráfico utiliza los componentes creados.

## U.T.5. Introducción a Windows Forms.

**RA1. Genera interfaces gráficos de usuario mediante editores visuales utilizando las funcionalidades del editor y adaptando el código generado.**

**RA3. Crea componentes visuales valorando y empleando herramientas específicas.**

**RA4. Diseña interfaces gráficos identificando y aplicando criterios de usabilidad.**

### Objetivos de aprendizaje:

- Crear un formulario y añadir controles.
- Crear un formulario heredado utilizando herencia visual.
- Organizar controles en un formulario.
- Crear aplicaciones MDI (Multiple Document Interface).

## Contenidos

- Crear un formulario
  - Windows Forms frente a Web Forms
  - Cómo crear un formulario
  - Cómo establecer las propiedades de un formulario
  - Ciclo de vida de un formulario
  - Cómo gestionar eventos del formulario
  - Código generado por el Diseñador de Windows Forms
- Añadir controles a un formulario
  - Cómo añadir controles a un formulario
  - Cómo añadir menús a un formulario
  - Cómo personalizar los controles del Cuadro de herramientas
- Crear un formulario heredado
  - Modificadores de acceso
  - Cómo crear un formulario heredado
- Organizar controles en un formulario
  - Cómo organizar controles en un formulario utilizando el menú Formato
  - Cómo establecer el orden de tabulación de los controles
  - Cómo delimitar (Anchor) un control en Windows Forms
  - Cómo acoplar (Dock) un control en Windows Forms
- Crear aplicaciones MDI
  - Aplicaciones SDI frente a aplicaciones MDI
  - Cómo crear aplicaciones MDI
  - Cómo interactúan los formularios principal y secundario

## Actividades de enseñanza-aprendizaje

- Evaluación inicial de conocimientos previos.
- Creación de aplicaciones con más de un formulario y menus
- Crear formulario heredado
- Organizar controles en formulario
- Crear aplicación MDI

## Criterios de evaluación

- Se ha creado un interfaz gráfico utilizando los asistentes de un editor visual.
- Se han utilizado las funciones del editor para ubicar los componentes del interfaz.
- Se han modificado las propiedades de los componentes para adecuarlas a las necesidades de la aplicación.
- Se ha analizado el código generado por el editor visual.
- Se han asociado a los eventos las acciones correspondientes.
- Se ha modificado el código generado por el editor visual.
- Se ha desarrollado una aplicación que incluye el interfaz gráfico obtenido.
- Se han identificado las herramientas para diseño y prueba de componentes.
- Se han creado componentes visuales.
- Se han definido sus propiedades y asignado valores por defecto.

- Se han determinado los eventos a los que debe responder el componente y se les han asociado las acciones correspondientes.
- Se han programado aplicaciones cuyo interfaz gráfico utiliza los componentes creados.
- Se han creado menús que se ajustan a los estándares.
- Se han creado menús contextuales cuya estructura y contenido siguen los estándares establecidos.
- Se han distribuido las acciones en menús, barras de herramientas, botones de comando, entre otros, siguiendo un criterio coherente.
- Se han distribuido adecuadamente los controles en la interfaz de usuario.
- Se ha utilizado el tipo de control más apropiado en cada caso
- Se ha diseñado el aspecto de la interfaz de usuario (colores y fuentes entre otros) atendiendo a su legibilidad.
- Se ha verificado que los mensajes generados por la aplicación son adecuados en extensión y claridad.
- Se han realizado pruebas para evaluar la usabilidad de la aplicación.

#### **U.T.6. Trabajar con controles.**

**RA1. Genera interfaces gráficos de usuario mediante editores visuales utilizando las funcionalidades del editor y adaptando el código generado.**

**RA3. Crea componentes visuales valorando y empleando herramientas específicas.**

**RA4. Diseña interfaces gráficos identificando y aplicando criterios de usabilidad.**

**Objetivos de aprendizaje:**

- Crear un controlador de eventos para un control.
- Seleccionar y utilizar los controles adecuados en una aplicación Windows Forms.
- Utilizar cuadros de diálogo en una aplicación Windows Forms.
- Añadir controles a un formulario en tiempo de ejecución.
- Crear y utilizar menús en una aplicación Windows Forms.
- Validar entrada de datos del usuario en una aplicación Windows Forms.

#### **Contenidos**

- Crear un controlador de eventos para un control.
  - Modelo de eventos del .NET Framework
  - ¿Qué es un controlador de eventos?
  - La palabra clave Handles
  - Cómo crear controladores de eventos para eventos de control
  - Cómo añadir y eliminar controladores de eventos en tiempo de ejecución

- Uso de los controles de los formularios Windows Forms.
  - Seleccionar un control de un formulario Windows Forms basándose en la función
  - Cómo utilizar el control StatusBar
  - Cómo utilizar el control ListBox
  - Cómo utilizar los controles GroupBox y Panel
  - Cómo utilizar los controles ToolBar e ImageList
  - Demostración: implementar operaciones de arrastrar y soltar entre controles
- Uso de los cuadros de diálogo en una aplicación Windows Forms.
  - Seleccionar cuadros de diálogo en Visual Studio .NET
  - Cómo mostrar cuadros de diálogo en una aplicación
  - La propiedad DialogResult
  - Cómo utilizar la entrada en los cuadros de diálogo
  - Demostración: uso del control OpenFileDialog
- Agregar controles en tiempo de ejecución.
  - Colección de controles
  - Cómo agregar controles en tiempo de ejecución
  - Práctica: agregar y eliminar controles en tiempo de ejecución
- Crear menús.
  - Cómo agregar un menú contextual a un formulario
  - Cómo agregar elementos de menú en tiempo de ejecución
  - Cómo crear controladores de menú para elementos de menú
  - Cómo utilizar propiedades de menú
- Validar introducción de datos de los usuarios.
  - Cómo validar controles utilizando el evento Validating
  - Control ErrorProvider
  - Cómo utilizar el control ErrorProvider
  - Demostración: validar datos en una aplicación Windows Forms

### **Actividades de enseñanza-aprendizaje**

- Evaluación inicial de conocimientos previos.
- Crear un controlador de eventos para un control

- Crear y utilizar un control ToolBar
- Implementar operaciones de arrastrar y soltar entre controles
- Uso de controles CommonDialog
- Agregar y eliminar controles en tiempo de ejecución
- Actualizar menús en tiempo de ejecución
- Validar datos en una aplicación

### **Criterios de evaluación**

- Se ha creado un interfaz gráfico utilizando los asistentes de un editor visual.
- Se han utilizado las funciones del editor para ubicar los componentes del interfaz.
- Se han modificado las propiedades de los componentes para adecuarlas a las necesidades de la aplicación.
- Se ha analizado el código generado por el editor visual.
- Se han asociado a los eventos las acciones correspondientes.
- Se ha modificado el código generado por el editor visual.
- Se ha desarrollado una aplicación que incluye el interfaz gráfico obtenido.
- Se han identificado las herramientas para diseño y prueba de componentes.
- Se han creado componentes visuales.
- Se han definido sus propiedades y asignado valores por defecto.
- Se han determinado los eventos a los que debe responder el componente y se les han asociado las acciones correspondientes.
- Se han realizado pruebas unitarias sobre los componentes desarrollados.
- Se han programado aplicaciones cuyo interfaz gráfico utiliza los componentes creados.
- Se han creado menús que se ajustan a los estándares.
- Se han creado menús contextuales cuya estructura y contenido siguen los estándares establecidos.
- Se han distribuido las acciones en menús, barras de herramientas, botones de comando, entre otros, siguiendo un criterio coherente.
- Se han distribuido adecuadamente los controles en la interfaz de usuario.
- Se ha utilizado el tipo de control más apropiado en cada caso
- Se ha diseñado el aspecto de la interfaz de usuario (colores y fuentes entre otros) atendiendo a su legibilidad.
- Se ha verificado que los mensajes generados por la aplicación son adecuados en extensión y claridad.
- Se han realizado pruebas para evaluar la usabilidad de la aplicación.

## U.T.7. Manejo de errores y excepciones.

### RA8. Evalúa el funcionamiento de aplicaciones diseñando y ejecutando pruebas.

#### Objetivos de aprendizaje:

- Definir y ver ejemplos de errores de sintaxis, tiempo de ejecución y lógica.
- Depurar código utilizando el depurador de Visual Studio .NET.
- Implementar en nuestra aplicación la gestión estructurada de excepciones.

#### Contenidos

- Tipos de errores.
  - ¿Qué son los errores de sintaxis?
  - ¿Qué son los errores de tiempo de ejecución?
  - ¿Qué son los errores de lógica?
  - Demostración: Tipos de errores
- Uso del depurador.
  - Demostración: cómo depurar una aplicación
  - ¿Qué es el modo de interrupción?
  - Cómo utilizar puntos de interrupción
  - Cómo modificar puntos de interrupción
  - La barra de herramientas **Depuración**
  - Recorrer paso a paso el código
  - Demostración: cómo utilizar las ventanas de depuración
  - Cómo utilizar las ventanas de depuración
  - Cómo utilizar la Ventana de comandos
- Gestión de excepciones.
  - La clase **Exception**
  - ¿Qué es la gestión estructurada de excepciones?
  - Cómo utilizar la instrucción **Try...Catch**
  - Cómo utilizar el bloque **Finally**
  - Cómo lanzar excepciones
  - Directrices para el uso de la gestión estructurada de excepciones
  - Demostración: uso de la gestión estructurada de excepciones

#### Actividades de enseñanza-aprendizaje

- Evaluación inicial de conocimientos previos.
- Diferenciar los tipos de errores
- Utilizar puntos de interrupción
- Utilizar las ventanas de depuración
- Utilizar la Ventana de comandos

- Lanzar excepciones
- Uso de la gestión estructurada de excepciones

### **Criterios de evaluación**

- Se ha establecido una estrategia de pruebas.
- Se han realizado pruebas de integración de los distintos elementos.
- Se han realizado pruebas de regresión.
- Se han realizado pruebas de seguridad.
- Se han realizado pruebas de uso de recursos por parte de la aplicación.
- Se ha documentado la estrategia de pruebas y los resultados obtenidos.

### **U.T. 8. Programación orientada a objetos en Visual Basic .NET.**

#### **RA3. Crea componentes visuales valorando y empleando herramientas específicas.**

#### **Objetivos de aprendizaje:**

- Entender los conceptos de programación orientada a objetos, incluyendo clase, abstracción, encapsulación y objeto.
- Utilizar el Examinador de objetos para examinar los elementos de programación disponibles, incluyendo clases y objetos.
- Crear una nueva clase, incluyendo sus métodos, propiedades y miembros de datos.
- Crear y utilizar la instancia de una clase, incluyendo miembros de instancia y de datos compartidos, y métodos compartidos y no compartidos.
- Explicar cómo funcionan los constructores y los destructores.
- Explicar la herencia, el polimorfismo y los espacios de nombres.

#### **Contenidos**

- Entender el concepto de clase
  - ¿Qué es una clase?
  - ¿Qué es un objeto?
  - Cómo utilizar el Examinador de objetos
- Trabajar con clases
  - Cómo crear una nueva clase
  - Cómo agregar miembros de datos de una instancia
  - Cómo agregar métodos
  - Cómo agregar propiedades
  - Cómo crear una instancia de una clase
  - Cómo utilizar los constructores
  - Cómo utilizar los destructores
- Uso de miembros compartidos



- Cómo utilizar los miembros de datos compartidos
- Cómo utilizar los métodos compartidos
- Herencia, polimorfismo y espacios de nombres
  - ¿Qué es la herencia?
  - Cómo heredar de una clase
  - ¿Qué es el polimorfismo?
  - Comparación de clases y estructuras
  - Cómo organizar clases en espacios de nombres

### Actividades de enseñanza-aprendizaje

- Evaluación inicial de conocimientos previos.
- Utilizar el Examinador de objetos
- Realizar aplicación con clases y objetos
- Crear métodos compartidos

### Criterios de evaluación

- Se han identificado las herramientas para diseño y prueba de componentes.
- Se han creado componentes visuales.
- Se han definido sus propiedades y asignado valores por defecto.
- Se han determinado los eventos a los que debe responder el componente y se les han asociado las acciones correspondientes.
- Se han realizado pruebas unitarias sobre los componentes desarrollados.
- Se han documentado los componentes creados.
- Se han empaquetado componentes.
- Se han programado aplicaciones cuyo interfaz gráfico utiliza los componentes creados.

### U.T.9. Acceso a datos con ADO.NET.

**RA2. Genera interfaces gráficos de usuario basados en XML utilizando herramientas específicas y adaptando el documento XML generado.**

**RA3. Crea componentes visuales valorando y empleando herramientas específicas.**

#### Objetivos de aprendizaje:

- Definir la terminología básica sobre bases de datos, incluyendo los conceptos de base de datos, tabla, registro, campo y clave.
- Describir algunos objetos ADO.NET utilizados habitualmente.
- Crear y abrir una conexión a una base de datos.
- Crear, leer, actualizar y eliminar registros de una base de datos.
- Utilizar el Asistente para formularios de datos para crear una sencilla aplicación de acceso a datos.

- Mostrar y modificar datos extraídos de una base de datos.

## Contenidos

- Conceptos de bases de datos
  - Terminología de las bases de datos
  - ¿Cómo funciona la programación de bases de datos?
  - ¿Qué es SQL?
  - ¿Qué es un entorno conectado?
  - ¿Qué es un entorno desconectado?
- Descripción de ADO.NET
  - ¿Qué es ADO.NET?
  - Objetos comunes de ADO.NET
  - Demostración del instructor: Uso del Explorador de servidores
  - Cómo trabajar con bases de datos en el Explorador de servidores
  - ADO.NET y XML
- Trabajar con datos
  - Cómo utilizar un objeto **Connection**
  - Cómo utilizar un objeto **DataAdapter**
  - Cómo utilizar un objeto **DataSet**
  - Cómo utilizar un control **DataGrid**
  - Cómo utilizar el Asistente para formularios de datos
- Acceder a datos con DataReaders
  - Crear un DataReader
  - Leer datos de un DataReader
  - Uso de DataReader frente a DataSet

## Actividades de enseñanza-aprendizaje

- Evaluación inicial de conocimientos previos.
- Utilizar el Explorador de servidores para visualizar elementos de las bases de datos.
- Identificar escenarios de datos conectados o desconectados
- Aplicación con acceso a datos
- Uso de asistente para formulario de datos

## Criterios de evaluación

- Se han identificado las herramientas para diseño y prueba de componentes.
- Se han creado componentes visuales.
- Se han definido sus propiedades y asignado valores por defecto.
- Se han determinado los eventos a los que debe responder el componente y se les han asociado las acciones correspondientes.
- Se han realizado pruebas unitarias sobre los componentes desarrollados.

- Se han documentado los componentes creados.
- Se han empaquetado componentes.
- Se han programado aplicaciones cuyo interfaz gráfico utiliza los componentes creados.
- Se han reconocido las ventajas de generar interfaces de usuario a partir de su descripción en XML.
- Se ha generado la descripción del interfaz en XML usando un editor gráfico.
- Se ha analizado el documento XML generado.
- Se ha modificado el documento XML.
- Se han asignado acciones a los eventos.
- Se ha generado el código correspondiente al interfaz a partir del documento XML.
- Se ha programado una aplicación que incluye el interfaz generado.

#### **U.T.10. Informes e impresión.**

#### **RA5. Crea informes evaluando y utilizando herramientas gráficas.**

##### **Objetivos de aprendizaje:**

- Crear informes utilizando las herramientas del entorno

##### **Contenidos**

- Utilización de la clase PrintDocument
- Impresión de archivos de texto de varias páginas
- Agregar cuadros de diálogo de Vista preliminar y Configurar página

##### **Actividades de enseñanza-aprendizaje**

- Evaluación inicial de conocimientos previos.
- Utilizar el Examinador de objetos
- Realizar aplicación con clases y objetos
- Crear métodos compartidos

##### **Criterios de evaluación**

- Se ha establecido la estructura del informe.
- Se han generado informes básicos a partir de una fuente de datos mediante asistentes.
- Se han establecido filtros sobre los valores a presentar en los informes.
- Se han incluido valores calculados, recuentos y totales.
- Se han incluido gráficos generados a partir de los datos.
- Se han utilizado herramientas para generar el código correspondiente a los informes de una aplicación.
- Se ha modificado el código correspondiente a los informes.
- Se ha desarrollado una aplicación que incluye informes incrustados.

## U.T.11. Distribución de aplicaciones.

### **RA7. Prepara aplicaciones para su distribución evaluando y utilizando herramientas específicas.**

#### **Objetivos de aprendizaje:**

- Distribuir y publicar aplicaciones

#### **Contenidos**

- Planificación y creación de una instalación distribuida
- Configuración de las opciones de instalación distribuida
- Elaborar y probar un proyecto de instalación distribuida
- Como examinar los archivos de instalación y como desinstalar

#### **Actividades de enseñanza-aprendizaje**

- Evaluación inicial de conocimientos previos.
- Distribuir una aplicación

#### **Criterios de evaluación**

- Se han empaquetado los componentes que requiere la aplicación.
- Se ha personalizado el asistente de instalación
- Se ha empaquetado la aplicación para ser instalada de forma típica, completa o personalizada.
- Se han generado paquetes de instalación utilizando el entorno de desarrollo.
- Se han generado paquetes de instalación utilizando herramientas externas.
- Se han generado paquetes instalables en modo desatendido.
- Se ha preparado el paquete de instalación para que la aplicación pueda ser correctamente desinstalada.
- Se ha preparado la aplicación para ser descargada desde un servidor web y ejecutada.

### 4.3. TEMPORALIZACIÓN DE LAS UNIDADES DE TRABAJO

En cuanto al tiempo asignado a cada UD, no es sencillo establecer una duración óptima y transferible a todas las situaciones. Dicha asignación depende mucho de la naturaleza de los módulos e incluso de la de cada UD. Un criterio que parece adecuado es que el logro, el resultado, al final de cada UD se promueva y se sitúa en un horizonte temporal cercano de entre 1 y 6 semanas. Este criterio de duración, establecida en semanas, permite diferentes adjudicaciones, en

función de la duración de cada módulo. En cualquier caso plantea un período de tiempo que permite unidades con suficiente carga horaria para desarrollar suficientemente cada UD. Este criterio de asignación horaria, como los demás, deben considerarse con la flexibilidad necesaria para poder contemplar e incorporar otros que no sean estrictamente pedagógicos, si bien éstos deberán ser considerados en primera instancia.

Así pues, permitir que cada UD establezca un logro con valor y significado propios y alcanzable en un plazo razonablemente cercano, puede considerarse que son dos aspectos que incorporan en el proceso de aprendizaje el elemento motivación, tan necesaria en todo proceso formativo.

Para este módulo, se establecen en el currículo para Comunidad Autónoma de la Región de Murcia, 120 horas, distribuidas en 6 horas semanales, dos trimestres, y equivalente a 9 ECTS.

### **Primer Trimestre (11 semanas, 60 horas)**

- U.T.0. Presentación del módulo (2 horas)
- U.T.1. Introducción a interfaz gráfica de usuario con VisualBasic.Net. (8 horas)
- U.T.2. Elementos del lenguaje. Variables y estructuras de datos. (8 horas)
- U.T.3. Funciones. Subrutinas y procedimientos. (8 horas)
- U.T.4. Bucles y estructuras de decisión. (8 horas)
- U.T.5. Introducción a Windows Forms. (10 horas)
- U.T.6. Trabajar con controles. (16 horas)

### **Segundo Trimestre (11 semanas, 60 horas)**

- U.T.7. Manejo de errores y excepciones. (8 horas)
- U.T. 8. Programación orientada a objetos en VisualBasic.NET. (20 horas)
- U.T.9. Acceso a datos con ADO.NET. (20 horas)
- U.T.10. Informes e impresión. (8 horas)
- U.T.11. Distribución de aplicaciones. (4 horas)

## 5. METODOLOGÍA

### 5.1. INTRODUCCIÓN.

Desarrollado en el punto **6.2.1** de la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.

### 5.2. ORIENTACIONES METODOLÓGICAS

Los aspectos metodológicos que se pretenden aplicar en este módulo descansan en la idea de que el alumno se considere parte activa de la actividad docente, con esto se pretende involucrarlo en el proceso de asimilación de nuevos conceptos y adquisición de capacidades no como un mero contenedor de éstas sino como un productor directo de estos conocimientos y habilidades en sí mismo. De igual forma se pretende que el alumno respete al profesor y a sus compañeros, respetando igualmente el material de la clase.

- La metodología didáctica favorecerá, mediante la integración de los contenidos científicos tecnológicos y organizativos una visión global y coordinada de los procesos productivos en los que debe intervenir el alumnado.
- El trabajo en el aula consistirá en la exposición oral de cada unidad de trabajo, para que, posteriormente, los alumnos desarrollen los ejercicios y prácticas propuestos.
- La metodología será participativa, favoreciendo el aprendizaje por descubrimiento. Partiendo de los conocimientos iniciales de los alumnos/as, estos deberán construir sus aprendizajes significativos.
- La metodología deberá ser eminentemente práctica, acompañada de situaciones que reflejen la realidad en la mayor medida posible, huyendo de ejemplos y ejercicios abstractos.
- Estos aspectos prácticos del módulo se desarrollarán en la forma de:
  - 1.Experiencias didácticas: las realiza el profesor.
  - 2.Experiencias prácticas: las realizan los alumnos, el profesor corrige técnicas de trabajo y evalúa resultados.
- Para el trabajo en el aula, los alumnos dispondrán de toda la documentación que se considere oportuna, además de la asistencia permanente del profesor.

### 5.3. ESTRATEGIAS DE ENSEÑANZA-APRENDIZAJE. TIPOS DE ACTIVIDADES.

Desarrollado en el punto **6.2.2** de la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.

### 5.4. ORGANIZACIÓN DEL AULA, ESPACIOS Y TIEMPOS

Desarrollado en el punto **7.3** de la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.

Organización tipo B.

### 5.5. MATERIALES CURRICULARES Y RECURSOS DIDÁCTICOS

Desarrollado en el punto **7.2** de la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.

## 6. CRITERIOS SOBRE EVALUACIÓN, PROCEDIMIENTOS E INSTRUMENTOS DE EVALUACIÓN

### 6.1. CRITERIOS DE EVALUACIÓN

La gestión de la enseñanza no universitaria por parte de la Comunidad Autónoma de la Región de Murcia, en virtud del Real Decreto 938/1999, de 4 de junio, por el que se traspasan funciones y servicios de la Administración del Estado, ha permitido la elaboración de normativa propia relativa a los principios básicos por los que deben regirse las normas de convivencia en los centros docentes. Se trata del Decreto 115/2005, de 21 de octubre, por el que se establecen las normas de convivencia en los centros docentes sostenidos con fondos públicos que imparten enseñanzas escolares. El precitado Decreto dispone en su artículo 13, el derecho de los alumnos a una evaluación basada en criterios objetivos y obliga a los centros a informar al alumnado y a sus padres o tutores acerca de los criterios de evaluación y calificación, así como de promoción y titulación. Asimismo, responsabiliza a los profesores, y en particular al tutor, del mantenimiento de una comunicación fluida a lo largo del curso, al objeto de informarles del aprovechamiento académico, calificaciones y evolución del proceso de aprendizaje.

La Orden de 1 de junio de 2006 (BORM 22 de julio), que desarrolla normativamente determinados aspectos del Decreto 115/2005, de 21 de octubre, regula en la Educación Secundaria Obligatoria, Bachillerato y Formación Profesional el derecho de los alumnos a ser evaluados conforme a criterios objetivos y establece las condiciones que garantizan dicha objetividad, a la vez que señala el procedimiento mediante el cual los alumnos, o sus representantes legales, pueden solicitar aclaraciones, o, en su caso, reclamar por escrito contra las decisiones y calificaciones que, se adopten, después de cada evaluación, y al finalizar el curso.

La citada Orden de 1 de junio, de acuerdo con el artículo 44 del Decreto 115/2005, la falta de asistencia a clase de modo reiterado puede provocar la imposibilidad de la aplicación correcta de los criterios de evaluación y la propia evaluación continua. El porcentaje de faltas de asistencia, justificadas e injustificadas, que originan la imposibilidad de aplicación de la evaluación continua se establece en el 30% del total de horas lectivas de la materia o módulo. Así mismo, el alumno que se vea implicado en esta situación se someterá a una evaluación extraordinaria.

Últimamente, el Real Decreto 1147/2011, de 29 de julio, por el que se establece la ordenación general de la formación profesional del sistema educativo establece en el "Artículo 51. Evaluación de las enseñanzas de formación profesional", que:

1. La evaluación del aprendizaje del alumnado de las enseñanzas de formación profesional se realizará por módulos profesionales. Los procesos de evaluación se adecuarán a las adaptaciones metodológicas de las que haya podido ser objeto el alumnado con discapacidad y se garantizará su accesibilidad a las pruebas de evaluación.

2. En todo caso, la evaluación se realizará tomando como referencia los objetivos, expresados en resultados de aprendizaje, y los criterios de evaluación de cada uno de los módulos profesionales, así como los objetivos generales del ciclo formativo o curso de especialización.

**Los criterios de evaluación agrupados para cada resultado de aprendizaje de este módulo se han detallado en el apartado 3.3. y 3.4. de la presente programación.**

## 6.2. PROCEDIMIENTOS E INSTRUMENTOS DE EVALUACIÓN DEL PROCESO DE APRENDIZAJE

Desarrollado en el punto **9.2.2.4.2** de la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.



Al tratarse de un módulo que versa sobre desarrollo de aplicaciones se aplicará el punto **9.2.2.4.2.2.2**

## 7. CRITERIOS DE CALIFICACIÓN

Hemos de distinguir claramente entre evaluación y calificación. La calificación está destinada a medir el grado de aprendizaje (cuantificar), respecto a la totalidad planteada. La evaluación que establece el nuevo sistema educativo, es fundamentalmente formativa con una doble función: proporcionar ayuda pedagógica, es decir, convertirla en instrumento didáctico que permita diagnosticar y corregir, y una segunda función como controladora del proceso educativo, determinando el grado del cumplimiento de las intenciones educativas.

### 7.1. INDICADORES DE VALORACIÓN

Desarrollado en el punto **9.2.2.4.3** de la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.

Este modulo seguirá el Método 1 de calificación, indicado en el punto **9.2.2.4.3.1.1**

Se establece la siguiente ponderación:

| N | Apartado  | Ponderación      |
|---|---|------------------|
| 1 | Actitud y comportamiento en el aula<br>Asistencias/puntualidad<br>Compañerismo<br>Participación<br>Trabajo diario | 5% <sup>1</sup>  |
| 2 | Ejercicios y prácticas basados en enunciado   | 15% <sup>2</sup> |
| 3 | Pruebas objetivas teórico-prácticas   | 80%              |

La calificación de cada evaluación será la media ponderada de todas las partes de la tabla anterior, siendo condición necesaria obtener un mínimo de 5 puntos sobre 10 en el apartado 3 (Pruebas objetivas teórico-prácticas).

La **calificación final** (punto 9.2.2.4.3.2.1, método 1) será calculada como la media aritmética ponderada de la obtenida en las evaluaciones parciales, ponderando la primera evaluación con un

<sup>1</sup> Este 5 % no se podrá aplicar a aquellos alumnos que tengan amonestaciones.

<sup>2</sup> Puede que este 15% parezca poco, y nos gustaría poder subir mucho este porcentaje, pero existe un problema fundamental, y es que dado el tipo de contenidos/procedimientos de este módulo, resulta bastante difícil garantizar la autoría de los ejercicios o prácticas realizados libremente por el alumno.

40 % y la segunda con un 60%, sean éstas de sesión ordinaria o de recuperación, siempre y cuando todas ellas tuvieran evaluación positiva. En cualquier otro caso la calificación será negativa.

En las sesiones de evaluación, tras estudiar y apreciar el aprovechamiento y conducta de cada alumno, se acordarán las medidas de ayuda y recuperación a seguir con aquellos alumnos que las necesiten.

Como criterio habitual existirá una única **prueba de recuperación** (punto 9.2.2.4.3.3.1, método 1) para cada evaluación, prueba que se podrá programar, a criterio del profesor, en los treinta días hábiles siguientes a las pruebas ordinarias o al final de la última evaluación. En el primer caso, a criterio del docente, podrá existir una segunda prueba de recuperación al final de la última evaluación.

Los objetivos, conceptos y criterios de evaluación y calificación serán exactamente los mismos en las pruebas de recuperación que en las ordinarias, pero dado que se trata de una "segunda o tercera oportunidad" se aplicará un coeficiente reductor de la calificación obtenida según el siguiente esquema:

Primera recuperación de una evaluación

| Calificación (c) Obtenida | Coeficiente reductor | Ejemplos de calificación inicial (ci) y final (cf) resultante              |
|---------------------------|----------------------|--|
| $c \geq 0$ AND $c \leq 5$ | 0                    | ci=2 -> cf=2; ci=4 -> cf=4; ci=5 -> cf=5                                   |
| $c > 5$ AND $c \leq 10$   | 60 % de $c - 5$      | ci=6 -> cf=6-(6-5)*3/5=5,4; ci=7 -> cf=7-(7-5)*3/5=5,8; cf=10-(10-5)*3/5=7 |

Segunda recuperación de una evaluación

| Calificación (c) Obtenida | Coeficiente reductor | Ejemplos de calificación inicial (ci) y final (cf) resultante              |
|---------------------------|----------------------|--|
| $c \geq 0$ AND $c \leq 5$ | 0                    | ci=2 -> cf=2; ci=4 -> cf=4; ci=5 -> cf=5                                   |
| $c > 5$ AND $c \leq 10$   | 80 % de $c - 5$      | ci=6 -> cf=6-(6-5)*4/5=5,2; ci=7 -> cf=7-(7-5)*4/5=5,4; cf=10-(10-5)*4/5=6 |

En el caso de que en turno de recuperación se obtenga una nota menor que la obtenida anteriormente, en recuperación o en turno ordinario, será el docente el que decidirá cuál de las calificaciones será la definitiva de ese período evaluativo.

**Actividades de recuperación para alumnos de 2º en el tercer trimestre (punto 9.2.2.4.3.4)**

## 7.2. CONVOCATORIAS ORDINARIAS Y EXTRAORDINARIAS.

Desarrollado en los puntos **9.2.2.1 al 9.2.2.3** de la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.

## 8. MEDIDAS DE ATENCIÓN A LA DIVERSIDAD

Desarrollado en el punto **11** de la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.

## 9. TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Uno de los elementos fundamentales en la educación está constituido por la implantación progresiva de las nuevas tecnologías de la información y comunicación. Su presencia va siendo cada vez más usual, cómo se puede observar, tanto por el volumen de servicios que ya se pueden realizar a través de las redes de comunicación como por los términos, siglas y palabras que se utilizan como www; e-mail, teletrabajo o "portal".

Las Nuevas tecnologías aplicadas a la educación se orientan hacia la utilización de los principales instrumentos audiovisuales, informáticos, multimedia, redes de comunicación y telecomunicaciones, con el fin de conseguir:

- La eliminación de las barreras espacio-temporales entre el profesor y el estudiante.
- Flexibilización de la enseñanza.
- Ampliación de la oferta educativa para el estudiante.
- Favorecer tanto el aprendizaje cooperativo, como el autoaprendizaje.
- Individualización de la enseñanza.
- Potenciación del aprendizaje a lo largo de toda la vida.
- Interactividad e interconexión de los participantes en la oferta educativa.
- Adaptación de los medios y las necesidades y características de los sujetos.
- Ayudar a comunicarse e interactuar con su entorno a los sujetos con necesidades educativas

especiales.

Los recursos que las Tecnologías de la Información y la Comunicación ofrecen son los medios audiovisuales, televisión y video, la utilización de la videoconferencia, las aplicaciones didácticas que ofrecen los software informáticos, el uso de materiales multimedia interactivos, y las redes de comunicación. Es este último concepto, se incluyen las aplicaciones de Internet.

Las redes de comunicación, sobre todo Internet, se han conformado como nuevos espacios de comunicación interpersonal y de información compartida, proporcionando una amplia variedad de experiencias, información, materiales y posibilidades de comunicación.

En cuanto al desarrollo de las TIC, en la Región de Murcia, hay que hacer referencia a la Orden de 7 de noviembre de 2001, de la Consejería de Educación y Universidades por la que se regula la elaboración del Proyecto sobre las Tecnologías de la Información y la Comunicación para la adscripción al Proyecto PLUMIER, y la figura del responsable de medios informáticos de los centros docentes públicos no universitarios de la Región de Murcia. El objetivo fundamental del Proyecto Plumier, es la disposición y potenciación del uso de las TIC por parte de todos los sectores de la comunidad educativa, mediante la dotación y utilización de recursos informáticos. En cualquier caso, este módulo pertenece a la especialidad Informática, por lo que constituye su propio objeto de estudio y así mismo se utilizan como medio de aprendizaje.

## 10. TEMAS TRANSVERSALES

Desarrollado en el punto **14** de la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.

## 11. PREVENCIÓN DE RIESGOS LABORALES

Desarrollado en el punto **12** de la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.

## 12. ACTIVIDADES COMPLEMENTARIAS Y EXTRAESCOLARES

- Participación Olimpiada Informática.
- Participación en LanParty.
- Visita al SICARM.
- Visita a empresas de tecnología y desarrollo.
- Y todas las actividades organizadas por el departamento.

## 13. EVALUACIÓN DE LOS PROCESOS DE LA ENSEÑANZA Y DE LA PRÁCTICA DOCENTE

Desarrollado en el punto **9.3** de la parte general de la programación anual del ciclo DAM del Departamento de la Familia profesional de Informática.