



**PROGRAMA PLANEAMIENTO EDUCATIVO
DEPARTAMENTO DE DISEÑO Y DESARROLLO CURRICULAR**

		PROGRAMA			
		Código en SIPE	Descripción en SIPE		
TIPO DE CURSO		049	Educación Media Tecnológica		
PLAN		2004	2004		
SECTOR DE ESTUDIO		620	Informática		
ORIENTACIÓN		48S	Informática Énfasis Desarrollo y Soporte		
MODALIDAD		-	Presencial		
AÑO		3ro.	Tercero		
TRAYECTO		-	-		
SEMESTRE		-	-		
MÓDULO		-	-		
ÁREA DE ASIGNATURA		915	Técnicas Informáticas Nivel II		
ASIGNATURA		35021	Programación III		
ESPACIO o COMPONENTE CURRICULAR		Tecnológico			
DURACIÓN DEL CURSO		Horas totales: 96	Horas semanales: 3		Cantidad de semanas: 32
Fecha de Presentación: 14/09/2018	Nº Resolución del CETP	Exp. Nº	Res. Nº	Acta Nº	Fecha __/__/__

FUNDAMENTACIÓN

.NET es una nueva tecnología que está siendo aceptada e implantada por la industria a un ritmo bastante rápido. Las posibilidades que ofrece esta nueva arquitectura,, junto con los aspectos más científicos que engloba, constituyen una interesante materia de estudio sobre la cual se puede realizar varias líneas de especialización, atendiendo a las prioridades de cada cual.

El objetivo principal de este curso es el de aprender a explotar las posibilidades de esta nueva plataforma, mediante el uso de un nuevo lenguaje de reciente aparición, C#, con el cual fue desarrollada y con el cual puede utilizarse todos los servicios que la plataforma pone a disposición del programador. El curso abarca desde la Programación Orientada al Objeto, hasta las más mínimas particularidades del lenguaje C#.

OBJETIVOS

El objetivo principal de la asignatura es formar al estudiante en el desarrollo de Sistemas Empresariales de porte mediano a grande en la plataforma .NET de Microsoft, poniendo foco en actividades de diseño de arquitectura y programación.

Objetivos Generales:

-) Introducir al estudiante en el diseño y desarrollo de sistemas de información en la plataforma NET de Microsoft
-) Entrenar en el estudiante las capacidades de autoestudio de tecnologías, y las habilidades de resolución de problemas técnicos
-) Fomentar en el estudiante habilidades de trabajo en equipo, y la colaboración entre grupos de trabajo
-) Entrenar las capacidades de comunicación y presentación oral del trabajo realizado

Objetivos Específicos:

-) Dominar los aspectos más importantes que describen el funcionamiento de la plataforma .NET y las principales características de su Interfaz de Desarrollo.
-) Crear y gestionar un proyecto .NET y los aspectos y más destacables y convenciones del lenguaje C#.

-) Manipulación de datos, variables y demás elementos necesarios para el diseño de sentencias y estructuras de control en C#.
-) Conocer las estructuras de iteración y automatización de tareas en el código C#.
-) Dominar los fundamentos de la Programación orientada a objetos y su implementación en C#.
-) Utilizar las funciones más destacadas para la manipulación de objetos.
-) Desarrollar el diseño de un entorno para usuario de una aplicación.
-) Conocer la inserción creación y gestión de controles para una interfaz de usuario.
-) Aprender el acceso con C# a bases de datos y el control del flujo de datos con la aplicación.
-) Conocer el empaquetado y generación de ejecutables para la distribución de un programa C#.

CONTENIDOS

1. Introducción a .NET Framework

Marzo-Abril

Introducción a .NET Framework

Tipos de datos .NET

Declaración y asignación de variables

Constantes

Conversiones entre tipos de datos

Operadores

-) Operadores aritméticos
-) Operadores de comparación
-) Operadores lógicos
-) Asignación condicional

Condicionales (toma de decisiones)

-) Instrucción If
-) Instrucción Switch/Select Case

Repeticiones (iteraciones)

-) Instrucción For
-) Instrucción While
-) Instrucción Do while/ Do Loop While

Métodos (procedimientos y funciones)

-) Características
-) Procedimientos
-) Funciones
-) Parámetros opcionales
-) Parámetros por valor y parámetros por referencia
-) Uso de enumeraciones

2. Arquitectura en tres capas

Abril

Capa de presentación

Capa de negocio

Capa de datos

Diferencia entre capas y niveles

3. Introducción a la programación orientada a objetos

Abril

Clases y objetos

-) Creación de clases
-) Instanciación de objetos

Atributos de una clase

-) Propiedades
-) Propiedades autoimplementadas
-) Propiedades de sólo lectura
-) Propiedades de sólo escritura

Modificadores de visibilidad

Modificadores this/Me

Constructores

Herencia

Clases abstractas

Clases selladas

4. Windows Forms

Mayo-Junio

Eventos

-) Eventos por defecto
-) Otros eventos

Clase Form

-) Propiedades de uso frecuente
-) Métodos de uso frecuente
-) Eventos de uso frecuente
-) Cómo abrir un formulario desde otro formulario

Algunos controles de uso frecuente

-) Label
-) TextBox
-) MaskedTextBox
-) NumericUpDown
-) RadioButton/GroupBox
-) CheckBox
-) Button
-) ComboBox
-) PictureBox
-) Panel
-) TabControl
-) MenuStrip
-) ContextMenuStrip
-) ToolStripContainer
-) ToolStrip
-) StatusBar

-) ProgressBar
-) DataGridView
-) TreeView
-) ListView
-) OpenFileDialog
-) SaveFileDialog
-) ColorDialog
-) WebBrowser

MessageBox

Settings

Formularios MDI

5. Colecciones

Junio

Arrays multidimensionales (vectores, matrices, cubos, etc.)

Clase ArrayList

-) Declaración
-) Recorrer sus items
-) Agregar items
-) Insertar items
-) Eliminar items
-) Saber si un item está contenido en la lista
-) Ordenar en forma ascendente
-) Buscar y obtener el índice de un elemento en la lista

List<t>/List(Of t)

-) Declaración
-) Recorrer sus items

SortedList<k, v>

-) Declaración
-) Agregar items
-) Acceder a un valor través de su clave

) Recorrer sus items

Queue<t>

) Declaración

) Agregar items

) Recorrer y vaciar sus items

Stack<t>

) Declaración

) Agregar items

) Recorrer y vaciar sus ítems

6. Relaciones entre objetos

Julio

Relación jerárquica

Relaciones de contención

Clases estáticas

) Miembros de instancia y miembros de clase

Interfaces

) Implementación de interfases

7. Administración de errores (excepciones)

Julio

Try/Catch/Finally

Filtrado de excepciones

Uso de excepciones para validar reglas de negocio

Buenas prácticas para la administración de excepciones

8. Depuración de código (debug)

Julio

Puntos de interrupción (breakpoints)

Ventanas de inspección (watch windows)

Ventana inmediata (immediate windows)

Control de versiones (GIT)

-) Instalación
-) Comandos
-) Flujos de trabajo
-) Múltiples entornos
-) GitHub

9. ADO.NET

Agosto-Octubre

Bases de datos soportadas por ADO.NET

ADO conectado y desconectado

Connection

-) Cadena de conexión
-) Conectar a una base de datos

Command

Parameter

DataReader

Transaction

-) Ejecutar una sentencia SQL del tipo escalar
-) Ejecutar una sentencia SQL que devuelve la cantidad de registros afectados
-) Ejecutar una sentencia SQL que devuelve un conjunto de registros
-) Ejecutar un procedimiento almacenado que devuelve un conjunto de registros
-) Ejecutar sentencias SQL en el contexto de una transacción

DataSet

DataSet asociados a bases de datos y a otros orígenes de datos

Diseñador de DataSets

Configurar opciones del TableAdapter

Agregar métodos al TableAdapter

METODOLOGÍA

La materia gira en torno a un proyecto propuesto por el docente para ser desarrollado en equipos de aproximadamente 4 estudiantes. Se dictarán 3 horas semanales de clases teórico-prácticas pero adicionalmente cada alumno deberá dedicar un promedio de 6 horas semanales para el estudio y realización de ejercicios prácticos.

En la segunda mitad del semestre, el estudiante deberá dedicar un promedio de 12 horas semanales para la realización del laboratorio. En este periodo los grupos de trabajo además de aprender los contenidos en clase, irán profundizando el Proyecto Final y sobre el final de la clase harán consultas al docente sobre las dudas que irán surgiendo.

Debido a que el curso exige la realización de un Proyecto Final, se trabajará coordinadamente con otras asignaturas y de acuerdo a los intereses académicos de los alumnos, contribuyendo a que el aprendizaje sea significativo.

Se trabajará sobre ejemplos y ejercicios basados en la realidad cercana del alumno. Los primeros trabajos se construirán en forma sencilla y se les agregarán progresivamente los elementos que se vayan tratando, permitiendo al estudiante la comprobación efectiva de sus avances en el dominio del tema y la utilización de los diferentes componentes.

El docente podrá proponer situaciones complejas en las cuales sea necesario trabajar en equipo para optimizar la tarea, vivenciando el enriquecimiento del trabajo colectivo frente al individual.

El alumno deberá presentar a fin de año un trabajo completo en formato de Proyecto Final que será evaluado por todos los docentes del curso.

La metodología de enseñanza debe centrarse en los contenidos del curso, buscando desarrollar competencias digitales y, de manera integrada, promover que los alumnos sean capaces de expresarse correctamente de forma oral, presentando en público sus creaciones y propuestas, comunicarse con sus compañeros de manera respetuosa y cordial, redactar documentación; profundizar en la resolución de problemas mediante el pensamiento computacional y con el uso de aplicaciones informáticas; aprender a aprender en un ámbito de conocimiento en continuo proceso de cambio que fomenta el desarrollo de estrategias de meta- aprendizaje; trabajar individualmente y en equipo de manera autónoma, construyendo y compartiendo el conocimiento, llegando a acuerdos sobre las responsabilidades propias y las de sus compañeros; tomar decisiones, planificar, organizar el trabajo y evaluar los resultados; crear contenido digital, entendiendo las posibilidades que ofrece como una forma de expresión personal y cultural, y de usarlo de forma segura y responsable.

EVALUACIÓN

La evaluación educativa se concibe como un proceso enfocado a la valoración del grado de logro alcanzado de los objetivos planteados. Es entonces un proceso de valoración de cómo se van desarrollando los procesos de enseñanza y aprendizaje con el fin de mejorarlos en ambos sentidos: mejorar la tarea docente y facilitar el desarrollo de los aprendizajes.

Los criterios de evaluación se extraen de la justificación del curso, de los objetivos y contenidos del mismo. Es el profesor quien deberá concretarlos y adaptarlos, en función de las características de su alumnado, los medios disponibles y la metodología de enseñanza.

La evaluación del alumno debe tener las siguientes características:

-)] Integral: ya que se ha de tener en cuenta todos los aspectos de la personalidad del alumno (conceptos, procedimientos, actitudes, capacidades de relación y comunicación así como del desarrollo autónomo de cada estudiante.).
-)] Continua: durante el proceso. Se ha de hacer una evaluación:
 - o Inicial: para detectar los conocimientos previos de los alumnos
 - o Progresiva: para saber en qué situación se encuentra el alumno en cada momento.
 - o Final: como síntesis de todo lo conseguido.
-)] Formativa: los resultados de la evaluación deben servir al alumno para mejorar sus procesos de aprendizaje con la orientación del profesor.

Los instrumentos de evaluación tienen por objeto dar información sobre el grado de adquisición de los aprendizajes esperados. La información recogida debe ser válida y fiable. La evaluación debe realizarse en múltiples ocasiones y a través de diferentes instrumentos como ser: observación, orales, trabajos domiciliarios (individuales o en grupo), trabajos en clase, escritos, prácticos, obligatorios, cuestionarios, exposiciones u otras.

En este curso la evaluación será individual y por equipo, ya que los estudiantes deben realizar un Proyecto Final integrado con todas las demás asignaturas.

El docente debe prever las entregas parciales, las defensas de las mismas y las devoluciones de los avances, hasta una Defensa Final que se realizará al final del curso y antes de la reunión final.

BIBLIOGRAFÍA

- J BRUNNER, J. (2003). Educación e Internet ¿la próxima revolución? Chile. Fondo de Cultura Económica.
- J COLL César (1994): Psicología y Curriculum. Paidós.
- J GIMENO SACRISTAN, J. (1998). El currículo: una reflexión sobre la práctica. Madrid: Morata
- J Deitel (2004). Instructor Manual for Simply C#. (Libro)
- J Programación orientada a objetos. Técnicas de programación. (Manual)
- J Andrew Stellman. Head First C#. 2nd Edition. (Libro)
- J “A programmer’s introduction to C#” escrito por Eric Gunnerson y publicado por Apress en 2000.
- J “C# and the .NET Framework”, escrito por Andrew Troelsen y publicado por Apress en 2001
- J “C# Essentials”, escrito por Beb Albahari, Peter Drayton y Brand Merrill y publicado por O’Reilly en 2000.
- J “C# Programming with the Public Beta”, escrito por Burton Harvey, Simon Robinson, Julian Templeman y Karli Watson y publicado por Wrox Press en 2000.
- J “Inside C#”, escrito por Tom Archer y publicado por Microsoft en 2000
- J “Presenting C#”, escrito por Christoph Wille y publicado por Sams Publishing en 2000.
- J “Professional C#”, escrito por Simon Robinson, Burt Harvey, Craig McQueen,
- J Christian Nagel, Morgan Skinner, Jay Glynn, Karli Watson, Ollie Cornes, Jerod Moemeka y publicado por Wrox Press en 2001.
- J “Programming C#”, escrito por Jesse Liberty y publicado por O’Reilly en 2001

PORTALES

“El Rincón en Español de C#” (<http://tdg.lsi.us.es/~csharp>), Ha sido desarrollado por profesores de la Facultad de Informática y Estadística de Sevilla,

“.NET Developers Center” (<http://www.msdn.microsoft.com/net>) de Microsoft, ya que al ser los creadores del C# y la plataforma .NET su información sobre los mismos suele ser la más amplia, fiable y actualizada.

Otros (en inglés):

- “C# Corner” (<http://www.c-sharpcorner.com>)
- “C# Help”(<http://www.csharp-help.com>)
- “C# Station” (<http://www.csharp-station.com>)
- “Codehound C#” (<http://www.codehound.com/csharp>)
- “csharpindex.com” (<http://www.csharpindex.com>)
- “Developersdex” (<http://www.developersdex.com/csharp>)
- “.NET Wire” (<http://www.dotnetwire.com>)