

PROGRAMA DE FORMACIÓN
TÉCNICO PROFESIONAL EN PROGRAMACIÓN DE SISTEMAS DE
INFORMACIÓN

MÓDULO
CONSTRUCCIÓN DE ELEMENTOS DE SOFTWARE 1

ALIANZA FUTURO DIGITAL
POLITÉCNICO COLOMBIANO JAIME ISAZA CADAVID
MEDELLÍN
FEBRERO DE 2008

Tabla de Contenido

1.	JUSTIFICACIÓN Y PERTINENCIA	3
2.	OBJETIVOS	3
2.1.	GENERAL	3
2.2.	ESPECÍFICOS	3
3.	REQUISITOS	¡Error! Marcador no definido.
4.	COMPETENCIAS RELACIONADAS	4
4.1.	COMPETENCIAS ESPECÍFICAS: NORMAS DE COMPETENCIA	4
4.2.	COMPETENCIAS TRANSVERSALES.....	4
4.3.	COMPETENCIAS BÁSICAS	¡Error! Marcador no definido.
5.	RESULTADOS DE APRENDIZAJE.....	5
6.	TABLA DE SABERES Y CRITERIOS Y EVIDENCIAS DE EVALUACIÓN	5
7.	CRÉDITOS Y HORAS Y JUSTIFICACIÓN DE LOS CRÉDITOS DE ACUERDO A LOS CONTENIDOS QUE SE DESARROLLAN	7
8.	PERFIL PROFESOR.....	8
8.1.	COMPETENCIAS TECNOLÓGICAS	8
8.2.	COMPETENCIAS PEDAGÓGICAS	8
8.3.	COMPETENCIAS ACTITUDINALES.....	8
9.	CONTENIDOS PROGRAMÁTICOS Y MECANISMOS DE EVALUACIÓN:.....	10
10.	AMBIENTES DE APRENDIZAJE REQUERIDOS:	13
11.	MEDIOS Y RECURSOS A UTILIZAR:.....	13
12.	BIBLIOGRAFÍA	13
13.	DATOS DE CONSTRUCCIÓN Y REVISIÓN DEL MÓDULO	13

1. JUSTIFICACIÓN Y PERTINENCIA

La formación de técnicos y tecnólogos en sistemas en especial aquellos cuyo perfil básico es el de desarrollo de software deben de ir involucrando paso a paso el conocimiento de herramientas de programación que le aporten a su conocimiento general y que le permitan desarrollar sus competencias, así como también responder a las necesidades del medio en su desempeño profesional. Por esto el módulo de construcción de Software se concibe como un elemento esencial del área informática que permite desarrollar competencias, básicas, transversales y específicas en todo el proceso de ciclo de vida del desarrollo de software muy especialmente en lo referente a la codificación utilizando un lenguaje de programación determinado.

Los lenguajes de programación ha venido evolucionando pasando por diferentes paradigmas y ambientes de desarrollo que van desde la programación estructurada, programación basada en eventos y llegando hasta la programación orientada a objetos en ambientes tipo consola, visual. Este crecimiento exige de personas capaces de enfrentar estos diferentes paradigmas y ambientes a través del conocimiento y aplicación de lenguajes de programación en los procesos de construcción de software.

2. OBJETIVOS

2.1. GENERAL

Proporcionar a los estudiantes los conceptos, elementos y estructuras básicas de un lenguaje de programación que permitan la construcción de elementos de software fundamentales requeridos dentro de la solución de un problema.

2.2. ESPECÍFICOS

2.2.1. Describir la interfaz y el entorno básico del lenguaje de programación.

2.2.2. Codificar aplicaciones que utilicen los mecanismos esenciales de la programación, como variables, constantes, operadores, expresiones y sentencias de control.

2.2.3. Escribir programas aplicando las estructuras de datos adecuadas a situaciones específicas.

2.2.4. Desarrollar software basado en los paradigmas actuales

3. REQUISITOS: CONOCIMIENTOS, SABERES, HABILIDADES Y EXPERIENCIAS PREVIAS

- ✓ Comunicación oral y escrita en la lengua nativa
- ✓ Aprendizaje y actualización permanentemente
- ✓ Capacidad de trabajo en equipo
- ✓ Capacidad de abstracción
- ✓ Conocimientos de informática relativos al ámbito de estudio
- ✓ Habilidades de búsqueda de información
- ✓ Habilidades informáticas básicas

4. COMPETENCIAS RELACIONADAS

4.1. COMPETENCIAS ESPECÍFICAS: PROVENIENTES DE LAS NORMAS DE COMPETENCIA

Norma de Competencia	Elementos de Competencia	Competencia	Nivel de desarrollo de la Competencia
Desarrollar el sistema que cumpla con los requerimientos de la solución informática.	Construir el Software para el sistema de acuerdo con la metodología de desarrollo seleccionada, la arquitectura y las especificaciones dadas por el cliente	Programar en un ambiente seleccionado	1. Conoce el entorno de trabajo del lenguaje seleccionado
			2. Maneja los fundamentos del lenguaje de programación
			3. Utiliza (codifica) las estructuras de programación en el lenguaje
			4. Emplea estructuras modulares de programación, código reusable y algoritmos distribuidos en el desarrollo de la solución del problema planteado

4.2. COMPETENCIAS BÁSICAS Y TRANSVERSALES

- ✓ Trabajar en equipo
- ✓ Demostrar trabajo en forma autónoma
- ✓ Desarrollar abstracción, análisis y síntesis
- ✓ Identificar plantear y solucionar problemas
- ✓ Interpretar textos técnicos en idioma extranjero
- ✓ Desarrollar capacidad de análisis lógico

5. RESULTADOS DE APRENDIZAJE

- ✓ Reconocer el entorno de trabajo de un lenguaje de programación para identificar su interfaz, forma de instalación, depuración y compilación, interactuando con el.
- ✓ Identificar los elementos básicos del lenguaje de programación para codificar la solución planteada a un problema real, mediante la utilización de datos y las operaciones con ellos, utilizando la sintaxis propia del lenguaje.
- ✓ Codificar algoritmos utilizando las estructuras de programación y el manejo de memoria estática aceptadas por el lenguaje para implementar la solución propuesta.
- ✓ Construir soluciones de software utilizando la modularidad del lenguaje para permitir la reusabilidad de código con el fin de optimizar el programa.

NOTA: Como requisito para que el estudiante logre estos aprendizajes debe identificar y analizar los saberes y actitudes referenciados en la siguiente tabla:

6. TABLA DE SABERES Y CRITERIOS Y EVIDENCIAS DE EVALUACIÓN

SABER HACER	SABER	SER	CRITERIOS Y EVIDENCIAS DE EVALUACIÓN
1. Interpretar el análisis y el diseño para desarrollar el sistema	Técnicas de especificación formal de análisis y diseño. Manuales de referencia de lenguajes programación y ambientes de desarrollo. Metodologías de desarrollo. Reconocer tipos de datos y operaciones	Crítico y analítico para realizar su labor. Autónomo y objetivo frente a las soluciones a desarrollar. Organizado y ético en el manejo de la información Creativo con visión sistémica y holística	Realiza la interpretación del análisis y el diseño del sistema a desarrollar Interpreta manuales de referencia de lenguajes programación

	con ellos		
2. Organizar la estructura del aplicativo de acuerdo a estándares internacionales y de la empresa.	Interpretar manuales técnicos del ambiente de desarrollo. Ambientes de desarrollo. Metodologías de desarrollo.	Cuidadoso y disciplinado para el desarrollo de su trabajo Analítico y ético para realizar su labor Creativo al organizar y presentar la información Respetuoso con las normas Hábil para realizar investigación	Asigna los nombres de los productos y recursos del sistema de acuerdo a los estándares de la empresa Interpreta correctamente las normas internacionales para desarrollo de software
3. Programar en el lenguaje seleccionado de acuerdo con el diseño y los estándares de programación definidos.	Características de lenguajes de programación Algoritmos	Ordenado y ético en el manejo de la información. Interesado en aprender nuevos conceptos y reevaluar los conceptos adquiridos previamente Receptivo y abierto para asimilar las normas	Construye en el lenguaje de programación de acuerdo al diseño establecido. Utiliza los estándares de denominación, nomenclatura de acuerdo a los estándares de la empresa Documenta los programas de acuerdo con la metodología de desarrollo y los estándares
4. Programar las interfaces de acuerdo con las especificaciones dadas	Conocimientos básicos de interfaces del programa.	Respetuoso y tolerante en el manejo de las relaciones interpersonales Interesado en interpretar las necesidades de trabajo e información de los demás Demuestra criterio argumentativo y honestidad Sencillo y claro para desarrollar su trabajo	Realiza interfaces para los programas de acuerdo con las especificaciones dadas
5. Realizar las pruebas unitarias, la verificación y	Aplicar metodologías de pruebas	Cuidadoso en la gestión de su trabajo	Hace listas de chequeo para probar y, verificar el programa.

validación del programa de acuerdo con los requerimientos del cliente y listas de chequeo.	Definir casos de estudio para pruebas Procedimientos de verificación y validación de programas Seguimiento de bitácoras de pruebas Documentación casos de prueba y de errores encontrados en las pruebas	Analítico y receptivo para realizar su labor. Argumentativo y honesto en sus juicios. Ordenado y critico en la planeación del trabajo Maneja análisis e inferencia Cuidadoso en la gestión de su trabajo Asume que el trabajo bajo estrés es posible para la gestión de su trabajo.	Realiza la validación del programa de acuerdo con los requerimientos del cliente
6. Establecer una línea base sobre la cual se va a comenzar el control de las versiones de los elementos desarrollados	Técnicas de versionamiento	Interesado en interpretar las necesidades de trabajo e información de los demás Cuidadoso en la gestión de su trabajo Hábil en la comunicación escrita Sencillo y claro para desarrollar su trabajo	Aplica técnicas de versionamiento a los programas elaborados

7. CRÉDITOS Y HORAS Y JUSTIFICACIÓN DE LOS CRÉDITOS DE ACUERDO A LOS CONTENIDOS QUE SE DESARROLLAN

Total Horas Módulo : Construcción de elementos de software					
	Tiempo Presencial		Tiempo Independiente	Tiempo Total	Créditos
	Tiempo Teórico	Tiempo Asistido			
1. Interpretar el análisis y el diseño para desarrollar el sistema	0	2	4		
2. Organizar la estructura del aplicativo de acuerdo a estándares internacionales y de la empresa.	0	2	4		
3. Programar en el lenguaje seleccionado de acuerdo con el diseño y los estándares de programación definidos.	4	14	38		
4. Programar las interfaces de acuerdo con las especificaciones dadas	1	3	8		
5. Realizar las pruebas unitarias, la verificación y validación del programa de acuerdo con los requerimientos del cliente y listas de chequeo.	1	3	8		

6. Establecer una línea base sobre la cual se va a comenzar el control de las versiones de los elementos desarrollados	1	1	2		
Total horas	7	25			
	32		64	96	2

El tiempo teórico se explica desde la experiencia de varios docentes en la temática del módulo, el tiempo asistido se refiere al trabajo que el estudiante requiere con acompañamiento del docente para desarrollar las competencias planteadas y el tiempo independiente es la validación de las competencias adquiridas en el tiempo teórico y el asistido.

7. PERFIL PROFESOR

7.1. COMPETENCIAS TECNOLÓGICAS

- ✓ Ingeniero de Sistemas o Informático preferiblemente con especialización en Ingeniería de Software o Desarrollo de software con experiencia laboral en el área, con habilidad para planear, diseñar y liderar proyectos utilizando diferentes estrategias que propicien una cultura de mejoramiento continuo.

7.2. COMPETENCIAS PEDAGÓGICAS

Para poder llevar a cabo las competencias pedagógicas debe tener experiencia docente, con habilidad para orientar y ejecutar la planeación y programación académica, de acuerdo con los lineamientos establecidos por la decanatura e institución.

Demostrar habilidad para implementar los lineamientos curriculares, en donde se apliquen metodologías, didácticas, investigación, recursos tecnológicos, y mecanismos de evaluación.

Emplear y enriquecer la metodología por proyectos, donde converjan los diferentes saberes en un proyecto integrador que apoye los perfiles del estudiante.

Demostrar capacidad para trabajar en equipos interdisciplinarios.

7.3. COMPETENCIAS ACTITUDINALES

Una formación humanística, ética y en valores, con sentido de pertenencia por la institución y que demuestre respeto por sus estudiantes y compañeros de trabajo.

Actitud coherente que fomente valores y favorezca conductas, hábitos y prácticas necesarias para lograr una mejor convivencia ciudadana dentro y fuera de la institución.

8. CONTENIDOS PROGRAMÁTICOS Y MECANISMOS DE EVALUACIÓN:

SECUENCIA LÓGICA NÚMERO 1: Dimensionar el problema					
Resultados de Aprendizaje	Saberes Hacer Relacionados	Estrategias de trabajo presencial	Estrategias de trabajo independiente	Criterios de evaluación	Tiempo
✓ Reconocer el entorno de trabajo de un lenguaje de programación para identificar su interfaz, forma de instalación, depuración y compilación, interactuando con el.	✓ Interpretar el análisis y el diseño para desarrollar el sistema ✓ Organizar la estructura del aplicativo de acuerdo a estándares internacionales y de la empresa.	El profesor: ✓ Expone el entorno de desarrollo y los conceptos básicos de un lenguaje de programación para la comprensión por parte de los estudiantes. ✓ Propone a los estudiantes lecturas sobre la estructura de un programa. ✓ Realiza actividades en las salas de cómputo que ejemplifican los procesos vistos en clase. ✓ Explica de forma personalizada las características de los trabajos a desarrollar. ✓ Asesora a los equipos de trabajo que desarrollan el proyecto integrador de semestre.	El estudiante: ✓ Consulta los conceptos básicos y resuelve los talleres propuestos por el profesor. ✓ Consulta sobre el entorno de trabajo del lenguaje seleccionado y genera un conjunto de apreciaciones y preguntas que le permitirán generar discusión en las clases. ✓ Expone sus dudas respecto a los talleres y al avance del caso de estudio ✓ Conformar grupos de trabajo donde cada uno debe compartir con los demás las soluciones planteadas a los talleres propuestos por el profesor ✓ Desarrolla el proyecto integrador de semestre con base en los criterios	✓ Realiza la interpretación del análisis y el diseño del sistema a desarrollar ✓ Interpreta manuales de referencia de lenguajes programación ✓ Asigna los nombres de los productos y recursos del sistema de acuerdo a los estándares de la empresa ✓ Interpreta correctamente las normas internacionales para desarrollo de software	TP 4 TI 8

			establecidos en clase.		
--	--	--	---------------------------	--	--

SECUENCIA LÓGICA NÚMERO 2: Codificar

Resultados de Aprendizaje	Saberes Hacer Relacionados	Estrategias de trabajo presencial	Estrategias de trabajo independiente	Criterios de evaluación	Tiempo
Codificar algoritmos utilizando las estructuras de programación y el manejo de memoria estática aceptada por el lenguaje para implementar la solución propuesta.	✓ Programar en el lenguaje seleccionado de acuerdo con el diseño y los estándares de programación definidos. ✓ Programar las interfaces de acuerdo con las especificaciones dadas	El profesor: ✓ Expone la forma de traducir el pseudo código al lenguaje de programación para la comprensión por parte de los estudiantes. ✓ Propone a los estudiantes lecturas sobre las sentencias de control. ✓ Realiza actividades en las salas de cómputo que ejemplifican los procesos vistos en clase. ✓ Explica de forma personalizada las características de los trabajos a desarrollar. ✓ Asesora a los equipos de trabajo que desarrollan el proyecto integrador de semestre.	El estudiante: ✓ Consulta las características del lenguaje y resuelve los talleres propuestos por el profesor. ✓ Consulta sobre la sintaxis de las sentencias y genera un conjunto de apreciaciones y preguntas que le permitirán generar discusión en las clases. ✓ Expone sus dudas respecto a los talleres y al avance del caso de estudio ✓ Conformar grupos de trabajo donde cada uno debe compartir con los demás las soluciones planteadas a los talleres propuestos por el profesor ✓ Desarrolla el proyecto integrador de semestre con	Construye en el lenguaje de programación de acuerdo al diseño establecido. Utiliza los estándares de denominación, nomenclatura de acuerdo a los estándares de la empresa Documenta los programas de acuerdo con la metodología de desarrollo y los estándares Realiza interfaces para los programas de acuerdo con las especificaciones dadas	TP 22 TI 46

			base en los criterios establecidos en clase.		
--	--	--	--	--	--

SECUENCIA LÓGICA NÚMERO 3: Validación y Control

Resultados de Aprendizaje	Saberes Hacer Relacionados	Estrategias de trabajo presencial	Estrategias de trabajo independiente	Criterios de evaluación	Tiempo
Construir soluciones de software utilizando la modularidad del lenguaje para permitir la reusabilidad de código con el fin de optimizar el programa.	✓ Realizar las pruebas unitarias, la verificación y validación del programa de acuerdo con los requerimientos del cliente y listas de chequeo. ✓ Establecer una línea base sobre la cual se va a comenzar el control de las versiones de los elementos desarrollados.	El profesor: ✓ Expone las pautas para probar y documentar programas. ✓ Propone a los estudiantes lecturas sobre las formas de chequear el funcionamiento del programa. ✓ Realiza actividades en las salas de cómputo que ejemplifican los procesos vistos en clase. ✓ Explica de forma personalizada las características de los trabajos a desarrollar. ✓ Asesora a los equipos de trabajo que desarrollan el proyecto integrador de semestre.	El estudiante: ✓ Consulta las reglas de las pruebas funcionales ✓ Expone sus dudas respecto a los talleres y al avance del caso de estudio ✓ Conformar grupos de trabajo donde cada uno debe compartir con los demás las soluciones planteadas a los talleres propuestos por el profesor ✓ Desarrolla el proyecto integrador de semestre con base en los criterios establecidos en clase.	✓ Hace listas de chequeo para probar y, verificar el programa. ✓ Realiza la validación del programa de acuerdo con los requerimientos del cliente ✓ Aplica técnicas de versionamiento a los programas elaborados	TP 6 TI 10

10. AMBIENTES DE APRENDIZAJE REQUERIDOS:

- ✓ Aula de clase dotada de mesas de trabajo
- ✓ Vivero del software
- ✓ Empresa
- ✓ Biblioteca
- ✓ Sala de computadores con software especializado para lenguajes de programación y acceso a Internet.

11. MEDIOS Y RECURSOS A UTILIZAR:

- ✓ Recursos: Bibliografía
- ✓ Medios: Computador, video beam Internet

12. BIBLIOGRAFÍA

- ✓ CEBALLOS, Fco. Javier. Lenguaje de programación C#. México: Alfaomega. 2002.
- ✓ DEITEL HARVEY M., DEITEL PAUL J, Como Programar C#, España, Pearson Prentice Hall, 2007
- ✓ FOXALL JAMES, Visual C# 2005, España, Anaya Multimedia, 2007
- ✓ JOYANES AGUILAR, Luis. Fundamentos de programación: algoritmos y estructuras de datos. Madrid: McGraw-Hill, 2004. 3ª.
- ✓ JOYANES AGUILAR, Luis; Programación Orientada a Objetos. Madrid: McGraw-Hill, 1998. 2ª ed
- ✓ MAYO JOSEPH, C# al Descubierto, España, Pearson Prentice Hall, 2002
- ✓ RODRIGUEZ GOMEZ-STERM MIGUEL, BESTEIRO GOROSTIZAGA MARCO ANTONIO, Desarrollo de Aplicaciones .NET con C#, España, Mcgraw-Hill, 2002

13. DATOS DE CONSTRUCCIÓN Y REVISIÓN DEL MÓDULO

Elaborado por: Fabio Alberto Vargas, Luis Emilio Velasquez y John Jairo Monsalve	Revisado por: John Jairo Monsalve, Fabio Vargas y Luis Emilio Velázquez
Institución : Politécnico Colombiano JIC, Tecnológico de Antioquia	Institución: Tecnológico de Antioquia, Politécnico Colombiano JIC
Fecha : Febrero de 2007	Fecha : Febrero de 2007
Revisado y actualizado por: Eugenia Mejía de Restrepo y Diana Patricia Tirado G. Asesoras metodológicas	Fecha : Febrero de 2007
Ultima revisión y actualización por: Diana Patricia Tirado G. Asesora metodológica	Fecha: Febrero 1 de 2008.
Aprobado por acta XX de Comité de Currículo	Fecha:
Versión N°: 1	