



GOBIERNO DE
MÉXICO

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



Programa de estudios del módulo

Creación de algoritmos

Currículum Laboral

Área:

Tecnología y transporte

Carrera:

Profesional Técnico-Bachiller en
Soporte y mantenimiento de equipo de cómputo

4º semestre

Editor: Colegio Nacional de Educación Profesional Técnica

Módulo: Creación de algoritmos

Área: Tecnología y transporte

Carrera: PT-B en Soporte y mantenimiento de equipo de cómputo

Semestre: cuarto

Horas por semana: 5

Fecha de diseño o actualización: 4 de noviembre de 2024.

Vigencia: a partir de la aprobación de la Junta Directiva y en tanto no se genere un documento que lo actualice.

© Colegio Nacional de Educación Profesional Técnica

Prohibida la reproducción total o parcial de esta obra por cualquier medio, sin autorización por escrito del CONALEP.

Directorio

Rodrigo Alejandro Rojas Navarrete

Dirección General

Hugo Nicolás Pérez González

Secretaría Académica

Patricia Alejandra Bernal Monzón

Dirección de Diseño Curricular

Creación de algoritmos

| Contenido | | Pág. |
|---------------------|--|------|
| Capítulo I: | Generalidades del Profesional Técnico-Bachiller | |
| 1.1 | Marco Curricular Común de la Educación Media Superior | 5 |
| 1.2 | Objetivo de la carrera | 6 |
| Capítulo II: | Aspectos específicos del módulo | |
| 2.1 | Presentación | 7 |
| 2.2 | Propósito del módulo | 8 |
| 2.3 | Mapa del módulo | 9 |
| 2.4 | Unidades de aprendizaje | 10 |
| 2.5 | Referencias | 17 |

CAPÍTULO I: Generalidades del Profesional Técnico-Bachiller

1.1 Marco Curricular Común de la Educación Media Superior

El Marco Curricular Común de la Educación Media Superior propone una apuesta curricular centrada en el desarrollo integral de las y los adolescentes y jóvenes, con la finalidad de formar estudiantes capaces de conducir su vida hacia su futuro con bienestar y satisfacción; con sentido de pertenencia social, conscientes de los problemas sociales, económicos y políticos que aquejan al país, dispuestos a participar de manera responsable y con toma de decisión hacia los procesos de la democracia participativa y compromiso por generar soluciones de las problemáticas que los aquejan y que tengan la capacidad de aprender a aprender en el trayecto de su vida. Que sean adolescentes y jóvenes capaces de erigirse como agentes de transformación social y que fomenten una cultura de paz y de respeto hacia la diversidad social, sexual, política y étnica; solidarios y empáticos.

A través del currículum laboral, el Profesional Técnico-Bachiller desarrollará competencias laborales extendidas pertinentes, buscando la transversalidad con los módulos del currículum fundamental y ampliado; permitiendo con ello desarrollar conocimientos, destrezas, habilidades, actitudes y valores que le permitan comprender los procesos productivos en los que está involucrado para enriquecerlos, transformarlos, resolver problemas, ejercer la toma de decisiones y desempeñarse en diferentes ambientes laborales, con una actitud creadora, crítica, responsable y propositiva; de la misma manera, fomenta el trabajo en equipo, colaborativo, el desarrollo pleno de su potencial en los ámbitos profesional, personal, así como la convivencia de manera armónica con el medio ambiente y la sociedad.

1.2 Objetivo de la carrera

PT-B en Soporte y mantenimiento de equipo de cómputo

Realizar los servicios de instalación, configuración, operación, mantenimiento y actualización de equipo, dispositivos periféricos, sistemas y redes de computadoras, incorporando tecnologías de vanguardia.

CAPÍTULO II: Aspectos específicos del módulo

2.1 Presentación

El módulo de **Creación de algoritmos** pertenece al currículum laboral extendido y se imparte en el cuarto semestre de la carrera de Profesional Técnico-Bachiller en **Soporte y mantenimiento de equipo de cómputo**, dentro del Trayecto Técnico Programación aplicada. Tiene como finalidad que la o el alumno adquiera las habilidades y destrezas necesarias para desarrollar algoritmos que dan solución a problemas, empleando elementos básicos de programación computacional.

Está conformado por dos unidades de aprendizaje. En la primera unidad se aborda el desarrollo de algoritmos como parte de la metodología aplicada en la solución de problemas por computadora. La segunda unidad, aborda la implementación de soluciones por computadora.

La contribución del módulo es desarrollar competencias profesionales esenciales para su perfil de egreso y para su inserción laboral, incluyendo conocimientos, destrezas, habilidades, actitudes y valores que se integran y relacionan con otros módulos del plan de estudios, como: Programación básica orientada a objetos y Programación básica en HTML.

La tarea educativa tendrá que diversificarse, a fin de que el personal docente realice funciones preceptoras, que consistirán en la guía y acompañamiento del alumnado durante su proceso de formación académica y personal y en la definición de estrategias de participación que le permitan incorporar a su familia en un esquema de corresponsabilidad que coadyuve a su desarrollo integral; por tal motivo, deberá destinar tiempo dentro de cada unidad para brindar este apoyo a la labor educativa de acuerdo con el Programa de Preceptorías.

Por otro lado, el alumnado deberá gestionar su aprendizaje, a fin de distribuir su tiempo para dedicar un porcentaje de la duración del módulo al estudio independiente, para reforzar el conocimiento previo o adquirido en clase, de tal forma que obtengan hábitos de estudio que le permitan ser autodidacta.

Finalmente, es necesario que al concluir cada resultado de aprendizaje se considere una sesión de clase en la cual se realice la recapitulación de los aprendizajes logrados, con el propósito de verificar que éstos se han alcanzado o, en caso contrario, determinar las acciones de mejora pertinentes. Cabe señalar que en esta sesión el alumno o la alumna que haya obtenido insuficiencia en sus actividades de evaluación o desee mejorar su resultado, tendrá la oportunidad de entregar nuevas evidencias.

2.2 Propósito del módulo

Crear algoritmos mediante el uso de lenguajes, estructuras y herramientas de programación para dar solución a problemas específicos.

2.3 Mapa del módulo

| Nombre del módulo | Unidad de aprendizaje | Resultado de aprendizaje |
|---|--|---|
| <p>Creación de algoritmos</p> <p>90 horas</p> | <p>1. Diseño de algoritmos.</p> <p>42 horas</p> | <p>1.1 Maneja los fundamentos y principios de la operación de computadoras y su utilidad en la resolución de problemas, utilizando secuencias de solución de acuerdo con los fundamentos de la algoritmia considerando sus alcances y limitaciones.</p> <p>22 horas</p> |
| | | <p>1.2 Diseña algoritmos que dan solución a problemas específicos, empleando las técnicas de diagrama de flujo y pseudolenguaje.</p> <p>20 horas</p> |
| | <p>2. Implementación de soluciones por computadora.</p> <p>48 horas</p> | <p>2.1 Utiliza los lenguajes de programación Pascal y C++, identificando sus elementos y componentes, en la codificación de algoritmos que dan solución a problemas específicos.</p> <p>28 horas</p> |
| | | <p>2.2 Implementa elementos complejos de algoritmos codificados, a partir de la aplicación de pruebas de escritorio.</p> <p>20 horas</p> |

2.4 Unidades de aprendizaje

| | | |
|----------------------------------|---|-----------------|
| Unidad de aprendizaje: | 1. Diseño de algoritmos. | 42 horas |
| Propósito de la unidad | Diseñar algoritmos aplicando la metodología y herramientas estandarizadas de acuerdo con la lógica de estructuración, para generar propuestas de solución a problemas específicos. | |
| Resultado de aprendizaje: | 1.1 Maneja los fundamentos y principios de la operación de computadoras y su utilidad en la resolución de problemas, utilizando secuencias de solución de acuerdo con los fundamentos de la algoritmia considerando sus alcances y limitaciones. | 22 horas |

| Actividades de evaluación | Evidencias por recopilar | Ponderación | Contenidos |
|--|--|-------------------|--|
| <p>1.1.1 Diseña una secuencia de solución de problemas por computadora especificando sus elementos básicos.</p> | <ul style="list-style-type: none"> • Secuencia de solución de problemas con elementos básicos. | <p>30%</p> | <p>A. Identificación de los componentes de la computadora.</p> <ul style="list-style-type: none"> • Hardware. • Software. • Firmware. <p>B. Manejo de aplicaciones de la computadora.</p> <ul style="list-style-type: none"> • Aplicaciones generales. <ul style="list-style-type: none"> - Procesador de textos. - Hoja de cálculo. - Base de datos. - Presentación de información. • Aplicaciones específicas. <ul style="list-style-type: none"> - Diseño gráfico. - Simulación. - Control. - Procesamiento. - Comunicación. • Programación y diseño de sistemas computacionales. <ul style="list-style-type: none"> - Sistemas operativos. - Lenguajes de programación. - Máquina. - Bajo nivel. - Alto nivel. <p>C. Manejo de fundamentos de la programación de computadoras.</p> |

| Actividades de evaluación | Evidencias por recopilar | Ponderación | Contenidos |
|---------------------------|--------------------------|-------------|--|
| | | | <ul style="list-style-type: none"> • Tipos de programación. <ul style="list-style-type: none"> - Secuencial. - Estructurada. - Orientada a objetos. - Lógica o de lenguaje natural. - Inteligencia artificial. • Programa. <ul style="list-style-type: none"> - Sección encabezado. - Sección cuerpo de programa. D. Manejo de las etapas de solución de problemas. <ul style="list-style-type: none"> • Análisis. (Caracterización del problema). • Solución general. (Uso de símbolos o lenguaje común). • Pruebas preliminares. (Pruebas de escritorio). • Implementación específica. (Codificación). • Pruebas de ajuste (Implementación). • Uso (Operación del programa). |

| | | |
|----------------------------------|--|-----------------|
| Resultado de aprendizaje: | 1.2 Diseña algoritmos que dan solución a problemas específicos, empleando las técnicas de diagrama de flujo y pseudolenguaje. | 20 horas |
|----------------------------------|--|-----------------|

| Actividades de evaluación | Evidencias por recopilar | Ponderación | Contenidos |
|--|--|-------------|---|
| 1.2.1 Diseña algoritmos que dan solución a problemas específicos descritos por el docente. | <ul style="list-style-type: none"> • Serie de 5 algoritmos, diseñados en diagrama de flujo y pseudocódigo. | 20% | <p>A. Desarrollo de diagramas de flujo.</p> <ul style="list-style-type: none"> • Símbolos utilizados en la construcción de diagramas de flujo. <ul style="list-style-type: none"> - Inicio y termino. - Proceso. - Toma de decisiones. - Entrada. - Salida. - Dirección de flujo. - Conector entre diagramas. - Conector dentro de un diagrama. • Resolución de ejercicios mediante diagramas de flujo. <p>B. Desarrollo de Pseudocódigos.</p> <ul style="list-style-type: none"> • Seudocódigo. • Seudolenguaje. <p>C. Manejo de reglas de programación.</p> <ul style="list-style-type: none"> • Diseño de algoritmos en etapas. • División del algoritmo en partes independientes. • Identificación de estructuras de datos. • Descripción general del algoritmo. • Verificación o la prueba de escritorio. |

Sesión para recapitulación y entrega de evidencias, al término de cada resultado de aprendizaje.

| | | |
|----------------------------------|--|-----------------|
| Unidad de aprendizaje: | 2. Implementación de soluciones por computadora. | 48 horas |
| Propósito de la unidad | Implementar soluciones por computadora, trasladando los algoritmos desarrollados en código de programación, considerando los lineamientos y la lógica del lenguaje C++, para dar solución a problemas específicos. | |
| Resultado de aprendizaje: | 2.1 Utiliza los lenguajes de programación Pascal y C++, identificando sus elementos y componentes, en la codificación de algoritmos que dan solución a problemas específicos. | 28 horas |

| Actividades de evaluación | Evidencias por recopilar | Ponderación | Contenidos |
|---|--|-------------|--|
| <p>2.1.1 Utiliza lenguajes de programación C++ y Turbo Pascal, identificando sus funciones y comandos principales para codificación de algoritmos.</p> | <ul style="list-style-type: none"> • Listado de funciones y comandos básicos para codificación. • Serie de 5 problemas específicos, codificados. | 30% | <p>A. Instalación de lenguajes de programación.</p> <ul style="list-style-type: none"> • Lenguaje C++. <ul style="list-style-type: none"> - Requerimientos técnicos. - Procedimiento de instalación. - Configuración. • Lenguaje Turbo Pascal. <ul style="list-style-type: none"> - Requerimientos técnicos. - Procedimiento de instalación. - Configuración. <p>B. Manejo de tipos de datos.</p> <ul style="list-style-type: none"> • Predefinidos por los lenguajes. <ul style="list-style-type: none"> - Enteros. - Decimales. - De cadena. - Boléanos. • Definidos por el usuario. <ul style="list-style-type: none"> - Subrango. - Enumerados. <p>C. Manejo de estructuras de datos:</p> <ul style="list-style-type: none"> • Estructuras estáticas <ul style="list-style-type: none"> - Variables de carácter. - Variables numéricas enteras. - Variables numéricas de punto flotante - Variables de tipo cadena (string). - Matrices (arrays). - Registros y archivos. • Estructuras dinámicas. |

| Actividades de evaluación | Evidencias por recopilar | Ponderación | Contenidos |
|---------------------------|--------------------------|-------------|--|
| | | | <ul style="list-style-type: none"> - Cola. - Pila. - Árbol. <p>D. Codificación de algoritmos en lenguaje turbo pascal.</p> <ul style="list-style-type: none"> • Datos enteros. • Datos reales. • Codificación de soluciones. <ul style="list-style-type: none"> - Análisis del problema. - Creación del algoritmo. - Pseudocódigo. <p>E. Codificación de algoritmos en lenguaje C++.</p> <ul style="list-style-type: none"> • Tipos básicos de datos. • Modificadores. • Tamaños y rango de bits. • Codificación de soluciones. <ul style="list-style-type: none"> - Análisis del problema. - Creación del algoritmo. - Pseudocódigo. - Codificación en lenguaje C++. <p>F. Diseño de la estructura de programas.</p> <ul style="list-style-type: none"> • Declaraciones. <ul style="list-style-type: none"> - Librerías de inclusión. - Declaración de funciones y/o procedimientos. - Definición de constantes y/o variables. • Estructuras de programación. <ul style="list-style-type: none"> - Asignación. - Decisión. - Cíclicas. - De selección múltiple. • Estructuras de datos. <ul style="list-style-type: none"> - Estáticas simples. |

| Actividades de evaluación | Evidencias por recopilar | Ponderación | Contenidos |
|---------------------------|--------------------------|-------------|--|
| | | | <ul style="list-style-type: none"> - Dinámicas. - Registros. - Arreglos. - Archivos. • Funciones. <ul style="list-style-type: none"> - Predefinidas por el lenguaje. - Definidas por el usuario. |

| | | |
|----------------------------------|--|-----------------|
| Resultado de aprendizaje: | 2.2 Implementa elementos complejos de algoritmos codificados, a partir de la aplicación de pruebas de escritorio. | 20 horas |
|----------------------------------|--|-----------------|

| Actividades de evaluación | Evidencias por recopilar | Ponderación | Contenidos |
|--|--|-------------|--|
| <p>2.2.1 Diseño y creación de algoritmos usando un lenguaje de programación para su codificación.</p> | <ul style="list-style-type: none"> • Codificación de 5 algoritmos con sus respectivas pruebas de escritorio. | 20% | <p>A. Verificación de estructuras de programación.</p> <ul style="list-style-type: none"> • Estructuras de asignación. <ul style="list-style-type: none"> - Expresiones simples. - Expresiones complejas. - Operadores. • Estructuras de decisión. <ul style="list-style-type: none"> - Estructuras sencillas. Operadores de relación. Operadores lógicos. - Estructuras de decisión anidadas. - Ejercicios aplicando estructuras de decisión. • Estructuras cíclicas. <ul style="list-style-type: none"> - Ciclos con control. Antes. Después. - Estructuras cíclicas anidadas. Estructura cíclica FOR-ENDFOR. • Estructuras de selección múltiple. • Pruebas de escritorio. <p>B. Verificación de funciones.</p> <ul style="list-style-type: none"> • Componentes de las funciones. <ul style="list-style-type: none"> - Especificador de tipo. - Nombre de la función - Declaración de parámetros. • Reglas de las funciones. • Argumentos de las funciones. |

Sesión para recapitulación y entrega de evidencias, al término de cada resultado de aprendizaje.

2.5 Referencias

Básicas:

- Adiego, J.y Ziviani, N. (2007). *Diseño de Algoritmos con Implementaciones en Pascal y C*. Ediciones Paraninfo.
- Ceballos, F. (2007). *C / C++ Curso de programación*. RA-MA Editorial.

Complementarias:

- Joyanes A. (2006). *Programación en Pascal*. McGraw-Hill.
- Joyanes A. (2006). *Programación en C++ algoritmos, estructuras de datos y objetos*. McGraw-Hill.
- Joyanes, A. (2008). *Fundamentos de programación*. McGraw-Hill.
- García, J., Fernández, J. y Calderón, A. (2002). *El lenguaje de programación C*. Pearson Educación.
- Algoritmia.net. (2003). *Introducción a los Algoritmos*. Consultado el 03 de octubre de 2024 en <http://www.algoritmia.net/articles.php?id=30>
- López, J. (2009). *Algoritmos y programación*. Consultado el 03 de octubre de 2024 en www.eduteka.org/pdfdir/AlgoritmosProgramacion.pdf
- Desarrolloweb.com. *Introducción a los algoritmos*. Consultado el 03 de octubre de 2024 en <http://www.desarrolloweb.com/articulos/2143.php>