**TÉCNICO SUPERIOR UNIVERSITARIO EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN**

**HOJA DE ASIGNATURA CON DESGLOSE DE UNIDADES TEMÁTICAS**

|  |  |
| --- | --- |
| 1. **Nombre de la asignatura** | Metodología de la Programación |
| 1. **Competencias** | Implementar aplicaciones de software; mediante técnicas de programación y considerando los requerimientos de la organización; para eficientar sus procesos. |
| 1. **Cuatrimestre** | Primero |
| 1. **Horas Prácticas** | 60 |
| 1. **Horas Teóricas** | 30 |
| 1. **Horas Totales** | 90 |
| 1. **Horas Totales por Semana Cuatrimestre** | 6 |
| 1. **Objetivo de la Asignatura** | El alumno diseñará algoritmos en pseudocódigo y diagramas de flujo, para resolver un problema determinado. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unidades Temáticas** | **Horas** | | |
| **Prácticas** | **Teóricas** | **Totales** |
| Conceptos básicos. | 10 | 8 | 18 |
| Expresiones | 8 | 4 | 12 |
| Algoritmos y Diagramas de Flujo | 42 | 18 | 60 |
| **Totales** | **60** | **30** | **90** |

# METODOLOGÍA DE LA PROGRAMACIÓN

**UNIDADES TEMÁTICAS**

|  |  |
| --- | --- |
| 1. **Unidad Temática** | I. Conceptos Básicos |
| 1. **Horas Prácticas** | 10 |
| 1. **Horas Teóricas** | 8 |
| 1. **Horas Totales** | 18 |
| 1. **Objetivo** | El alumno determinará las entradas, procesos y salidas de un problema, utilizando los elementos básicos de programación, para el planteamiento de una solución. |

| **Temas** | **Saber** | **Saber hacer** | **Ser** |
| --- | --- | --- | --- |
| Adquisición, procesamiento y almacenamiento de datos | Identificar los elementos básicos del proceso de información (entrada - proceso - salida) | Esquematizar el proceso de información. | Analítico.  Ordenado.  Sistemático. |
| Conceptos de Algoritmos, Pseudocódigo y Diagrama de Flujo | Reconocer la diferencia entre un algoritmo, pseudocódigo y diagrama de flujo | Determinar de los distintos tipos de técnicas la adecuada para la solución de problemas. | Analítico.  Sistemático  Ordenado  Asertivo. |
| Tipos de datos (numéricos, caracteres / cadenas, booleanos. | Identificar los diferentes tipos de datos utilizados en la programación, aplicándolo en la resolución de un problema. | Determinar el tipo de dato a utilizar dependiendo de los requerimientos de información a manipular. | Analítico.  Ordenado.  Sistemático. |
| Identificadores, variables y constantes | Reconocer los conceptos de identificador, variable y constante.  Describir las reglas de creación de identificadores. | Determinar las variables y constantes a utilizar, dependiendo del tipo de datos a emplear y los requerimientos. | Analítico.  Ordenado.  Sistemático |
| Metodología para la solución de problemas por medio de computadoras | Identificar las etapas de la metodología de solución de problemas por medio de computadoras. | Determinar las entradas, proceso y salidas de un problema. | Analítico.  Ordenado.  Sistemático.  Coherente.  Proactivo.  Asertivo  Hábil para trabajo en equipo. |

# METODOLOGÍA DE LA PROGRAMACIÓN

|  |  |  |
| --- | --- | --- |
| **Proceso de evaluación** | | |
| **Resultado de aprendizaje** | **Secuencia de aprendizaje** | **Instrumentos y tipos de reactivos** |
| El alumno realizará un documento con 15 ejercicios, elaborando para cada uno de ellos, una tabla en la que identifiquen:   * + los valores de entrada, procesos y salidas   + los tipos de datos a emplear asociados a las variables o constantes y   + las operaciones a realizar. | 1. Comprender el proceso de información: entrada, proceso y salida. 2. Identificar los tipos de datos que procesa la computadora. 3. Analizar los elementos de entrada, proceso y salida ante un problema planteado. 4. Plantear los elementos de entrada, proceso y salida ante un problema. | Ejercicios prácticos  Listas de cotejo |

# METODOLOGÍA DE LA PROGRAMACIÓN

|  |  |
| --- | --- |
| **Proceso enseñanza aprendizaje** | |
| **Métodos y técnicas de enseñanza** | **Medios y materiales didácticos** |
| Solución de problemas  Juegos (competencias entre equipos) | Pizarrón, plumones, computadora  Cañón,  Impresos (ejercicios prácticos) |

|  |  |  |
| --- | --- | --- |
| **Espacio Formativo** | | |
| **Aula** | **Laboratorio / Taller** | **Empresa** |
| **X** |  |  |

# METODOLOGÍA DE LA PROGRAMACIÓN

**UNIDADES TEMÁTICAS**

|  |  |
| --- | --- |
| **1. Unidad Temática** | II. Expresiones |
| **2. Horas Prácticas** | 8 |
| **3. Horas Teóricas** | 4 |
| **4. Horas Totales** | 12 |
| **5. Objetivo** | El alumno convertirá expresiones algebraicas, aritméticas y lógicas a expresiones algorítmicas, considerando la jerarquía de operadores, para realizar las tareas solicitadas. |

| **Temas** | **Saber** | **Saber hacer** | **Ser** |
| --- | --- | --- | --- |
| Operadores aritméticos, relacionales y lógicos | Identificar los tipos de operadores existentes. | Localizar los tipos de operadores | Analítico.  Ordenado.  Sistemático.  Coherente.  Asertivo |
| Jerarquía de operadores | Identificar la jerarquía de los operadores. | Resolver expresiones aritméticas y lógicas. | Analítico.  Ordenado.  Sistemático.  Coherente.  Proactivo.  Asertivo |
| Expresiones | Identificar las expresiones aritméticas y/o lógicas inmersas en el planteamiento de un problema. | Convertir expresiones algebraicas, aritméticas y lógicas a expresiones algorítmicas | Analítico.  Ordenado.  Sistemático.  Coherente.  Proactivo.  Asertivo  Hábil para trabajo en equipo. |

# METODOLOGÍA DE LA PROGRAMACIÓN.

|  |  |  |
| --- | --- | --- |
| **Proceso de evaluación** | | |
| **Resultado de aprendizaje** | **Secuencia de aprendizaje** | **Instrumentos y tipos de reactivos** |
| El alumno entregará un cuadernillo con al menos 45 ejercicios que contenga:   * Conversión de expresiones algebraicas, aritméticas y lógicas a expresiones algorítmicas (15 de cada una). * La jerarquía de operadores indicando el orden de ejecución de cada una de ellas. | 1. Identificar los operadores, su función y su jerarquía.  2. Comprender los operadores, su función y su jerarquía.  3. Analizar la expresión.  4. Convertir la expresión. | Ejercicios prácticos  Listas de cotejo |

# METODOLOGÍA DE LA PROGRAMACIÓN

|  |  |
| --- | --- |
| **Proceso enseñanza aprendizaje** | |
| **Métodos y técnicas de enseñanza** | **Medios y materiales didácticos** |
| Solución de problemas  Juegos (competencias entre equipos) | Pizarrón, plumones, computadora  cañón,  Impresos (ejercicios prácticos) |

|  |  |  |
| --- | --- | --- |
| **Espacio Formativo** | | |
| **Aula** | **Laboratorio / Taller** | **Empresa** |
| **X** |  |  |

**METODOLOGÍA DE LA PROGRAMACIÓN**

**UNIDADES TEMÁTICAS**

|  |  |
| --- | --- |
| **Unidad Temática** | III. Algoritmos y Diagramas de flujo. |
| **Horas Prácticas** | 42 |
| **Horas Teóricas** | 18 |
| **Horas Totales** | 60 |
| **Objetivo** | El alumno elaborará algoritmos, diagramas de flujo y pseudocódigos para la resolución de problemas. |

| **Temas** | **Saber** | **Saber hacer** | **Ser** |
| --- | --- | --- | --- |
| Reglas para elaborar algoritmos, Pseudocódigo y diagramas de flujo. | Reconocer la sintaxis utilizada para el desarrollo de pseudocódigo.  Identificar la simbología utilizada en el desarrollo de diagramas de flujo.  Describir el procedimiento para realizar una prueba de escritorio. | Plantear la solución de un problema empleando pseudocódigo y diagramas de flujo. | Analítico.  Ordenado.  Sistemático.  Coherente.  Proactivo.  Asertivo |
| Contadores y acumuladores. | Reconocer el funcionamiento de un contador y un acumulador. | Comparar entre un contador y un acumulador. | Analítico.  Ordenado.  Sistemático.  Coherente.  Proactivo.  Asertivo. |
| Estructuras de control de selección | Reconocer el funcionamiento de las estructuras de control selectivas SimplesCompuestas Múltiples | Resolver un problema determinado aplicando las estructuras de control selectivas y verificar la solución a través de una prueba de escritorio. | Analítico.  Ordenado.  Sistemático.  Coherente.  Proactivo.  Asertivo  Planificador.  Creativo  Hábil para trabajo en equipo. |
| Estructuras de repetición | Reconocer el funcionamiento de las estructuras de control repetitivas  a) Para-hasta  b) Repite  c) Mientras. | Resolver un problema determinado aplicando las estructuras de control repetitivas y verificar la solución a través de una prueba de escritorio. | Analítico.  Ordenado.  Sistemático.  Coherente.  Proactivo.  Asertivo  Planificador.  Creativo  Hábil para trabajo en equipo. |

**METODOLOGÍA DE LA PROGRAMACIÓN**

|  |  |  |
| --- | --- | --- |
| **Proceso de evaluación** | | |
| **Resultado de aprendizaje** | **Secuencia de aprendizaje** | **Instrumentos y tipos de reactivos** |
| El alumno realizará un cuadernillo con 60 ejercicios resueltos que contenga:   * + Estructuras de control,   + Diagramas de flujo,   + Pseudocódigo, y   + Pruebas de escritorio (mínimo dos corridas con valores distintos). | 1. Identificar la simbología utilizada en los diagramas de flujo y los criterios para pseudocódigo.  2. Comprender la simbología utilizada en los diagramas de flujo y los criterios para pseudocódigo.  3. Analizar el problema.  4. Desarrollar la solución en pseudocódigo y diagrama de flujo.  5. Realizar la prueba de escritorio. | Ejercicios prácticos  Listas de cotejo |

# METODOLOGÍA DE LA PROGRAMACIÓN

|  |  |
| --- | --- |
| **Proceso enseñanza aprendizaje** | |
| **Métodos y técnicas de enseñanza** | **Medios y materiales didácticos** |
| Solución de problemas  Análisis de casos | Pizarrón, plumones, computadora  cañón,  Impresos (ejercicios prácticos, casos), Software de diagramación, Pseudointérpretes (ALICE, KAREL, PSEINT, TURTLE, DFD, LOGO) |

|  |  |  |
| --- | --- | --- |
| **Espacio Formativo** | | |
| **Aula** | **Laboratorio / Taller** | **Empresa** |
|  | **X** |  |

**METODOLOGÍA DE LA PROGRAMACIÓN**

**CAPACIDADES DERIVADAS DE LAS COMPETENCIAS PROFESIONALES A LAS QUE CONTRIBUYE LA ASIGNATURA**

| **Capacidad** | **Criterios de Desempeño** |
| --- | --- |
| Determinar los requerimientos de funcionalidad del programa mediante técnicas de análisis; para determinar las variables de entrada y salida del problema específico mediante técnicas de análisis; para determinar las variables de entrada y salida del problema específico | 1. Determina el objetivo del programa. 2. Realiza un mapa conceptual del problema. 3. Genera un listado de variables E/S asociándoles un tipo de dato. |
| Proponer una solución mediante un algoritmo y/o diagrama de flujo, para atender los requisitos de un problema especifico. | 1. Establece los pasos lógicos a seguir y su orden. 2. Usa las estructuras de control necesarias. 3. Presenta el pseudocódigo del problema y/o diagrama de flujo correspondiente al modelado de objetos 4. Define un escenario de pruebas para evaluar que los resultados son acordes a los requerimientos. |
| Desarrollar la aplicación utilizando algún lenguaje de programación; para solucionar un problema especifico | 1. Realiza la traducción del diseño al lenguaje de programación. 2. Interpreta los errores de compilación y los corrige. 3. Generar un programa ejecutable y realiza la verificación de los resultados (errores lógicos) al menos con 2 corridas usando valores de entrada diferentes. 4. Presenta el código fuente del programa de acuerdo a las buenas prácticas de programación (tabulaciones, comentarios, nombres de variables, entre otras) |
| Diseñar la Base de Datos acorde a los requerimientos de procesamiento de información; para clasificar la información a procesar. | 1. Recolecta los requerimientos, clasificándolos dependiendo de las diferentes entidades, identificando los datos a manipular. 2. Genera un diagrama Entidad / Relación. 3. Realiza el modelo relacional del diagrama anterior y el esquema de la BD normalizada, empleando las 3 primeras Formas Normales, de acuerdo a las características de la base de datos. |
| Operar la Base de Datos mediante un manejador de bases de datos; para crear estructuras, insertar, borrar, modificar y extraer datos registrados. | * 1. Crea y/o modifica la estructura de la BD empleando SQL.   2. Genera sentencias, empleando SQL, para realizar inserciones, eliminaciones y modificaciones y presenta la base de datos con los cambios realizados.   3. Extrae información de la BD por medio de consultas en SQL.   4. Interpreta errores y los corrige. |

**METODOLOGÍA DE LA PROGRAMACIÓN**

**FUENTES BIBLIOGRÁFICAS**

| **Autor** | **Año** | **Título del Documento** | **Ciudad** | **País** | **Editorial** |
| --- | --- | --- | --- | --- | --- |
| Alcalde Eduardo, García Miguel | 1990 | Metodología de La Programación | Madrid | España | [Mc Graw-Hill Interamericana De España](http://dialnet.unirioja.es/servlet/editor?codigo=2558) |
| Baltasar García, J. | 2008 | ***Metodología de la Programación*** | Madrid | España | Prentice Hall |
| Cairo Battistuti Osvaldo | 2004 | *Metodología De La Programación* | D.F. | México | Alfaomega Grupo Editor S.A. De C.V. |
| Cairo Battistuti Osvaldo | 2008 | ***Metodología de La Programación: Algoritmos, Diagramas De Flujo Y Programas (3ª Ed.)*** | Buenos Aires | Argentina | Computec |
| Joyanes Aguilar Luis | 1988 | Metodología de la Programación. Diagramas de Flujo, Algoritmos Y Programación Estructurada | Madrid | España | Mc Graw Hill Interamericana Editores, S.A. De C.V. |
| Joyanes Aguilar Luis | 1992 | Problemas de Metodología de La Programación | Madrid | España | Mc Graw Hill Interamericana Editores, S.A. De C.V. |
| Katcheroff, Pablo | 2006 | *El Gran libro de la Programación* | Buenos Aires | Argentina | MP Ediciones |
| López Leobardo | 2006 | *Metodología De La Programación Orientada A Objetos* | D.F. | México | Alfaomega Grupo Editor S.A. De C.V. |
| Rodríguez Almeida Ángel M. | 1991 | *Metodología De La Programación a través de Pseudocódigo.* | Madrid | España | Mc Graw Hill Interamericana Editores, S.A. De C.V. |