



PROGRAMACIÓN

COMPETENCIA GENERAL

REALIZA PROGRAMAS COMPUTACIONALES BAJO UN ENFOQUE ESTRUCTURADO Y UN LENGUAJE DE PROGRAMACIÓN EN EL CAMPO DE LAS TELECOMUNICACIONES

COMPETENCIAS

PARTICULARES

1. Diseña algoritmos y diagramas de flujo para la solución de problemas técnicos y científicos para el área de telecomunicaciones

RAP 1: Elabora el algoritmo de la solución del problema empleando la técnica HIPO

RAP 2: Elabora diagramas de flujo de los algoritmos correspondientes de acuerdo a simbología estandarizada.

2. Codifica las soluciones obtenidas de los diagramas de flujo de acuerdo a las técnicas de la programación estructurada y gramática del lenguaje computacional.

RAP 1: Codifica diagramas de flujo de acuerdo a la sintaxis y semántica del lenguaje de programación.

RAP 2: Realiza programas empleando el enfoque de la programación estructurada

3. Desarrolla la solución de proyectos en el área de telecomunicaciones aplicando las técnicas de programación estructurada y paradigmas actuales de computación.

RAP 1: Estructura la solución de un proyecto empleando los arquetipos de la programación estructurada.

RAP 2: Efectúa la codificación y validación del proyecto de acuerdo a las normas del lenguaje computacional y el método de análisis de casos.



INTRODUCCIÓN

El NUEVO MODELO EDUCATIVO del INSTITUTO POLITÉCNICO NACIONAL y la globalización mundial, así como las nuevas tecnologías de la informática y computación, nos obliga a una constante actualización y revisión de los planes y programas de estudio, con la finalidad de incrementar el nivel de enseñanza para verse proyectada en la libre competitividad y vinculación con el sector productivo del país.

La Industria se encuentra en constante actualización e innovación y las competencias profesionales implican como principales objetos de conocimiento que permitirá que el estudiante resuelva el cumplimiento normativo vigente confirmando que el medio utilizado es el adecuado en función a los sistemas de telecomunicaciones.

Estas necesidades de actualización se tomaron en cuenta en el Nuevo Plan de Estudios y por consiguiente en el programa de estudios de la presente guía, como parte fundamental de formación de Técnicos en Telecomunicaciones.

JUSTIFICACIÓN.

Esta guía es una herramienta complementaria y de apoyo para preparar al estudiante en el desarrollo de diversas estrategias de estudio, que lo conozca los medios para facilitar su aprendizaje significativo, estratégico, autónomo, colaborativo, reflexivo, crítico y creativo.



En esta guía se establecen las actividades, cuestionarios y actividades adecuados para que los alumnos puedan desarrollar su aprendizaje ya sea en forma autónoma o con el apoyo y guía del profesor en el aula, pretendiendo ubicarlos adecuadamente en sus actividades extra-clase, o bien prepararlos convenientemente para realizar cualquier tipo de examen de la asignatura durante el periodo escolar.

METAS.



Que esta guía le sirva al alumno como herramienta practica de su aprendizaje y que con ella logre desarrollar y realizar programas computacionales bajo un enfoque estructurado y un lenguaje de programación en el campo de las telecomunicaciones.

EVALUACIÓN.

Los aprendizajes conceptuales son evaluados a través de la correcta resolución de cuestionarios, los aprendizajes procedimentales son evaluados a través de la resolución de actividades desarrolladas en los trabajos prácticos y de campo, estos trabajos serán evaluados al final de cada RAP por medio de listas de cotejo y guías de observación, dependiendo de las actividades desarrolladas, lo que permite que la evaluación sea continua y sumativa para el alumno.

```
Borland C++ for DOS
File Edit Search Run Compile Debug Project Options Window Help
[ ] ELC.C
/*
* File: ELC.C:-----
*
* The Endless Lands Games And Realities Compiler.
*
*-----
* Linking
*/
static char *PUCS
/*
* Revision Histor
*
* 1:1
*-----
*Compiling ELC.C:
Warning ELC.C 27:
Compiling LOGS.C:
Warning LOGS.C 114: 'PUCS' is declared but never used
Compiling LEXV.C:
-----
Linking
EXE file : ELC.EXE
Linking : \BORLANDC\LIB\CL.LIB
-----
Total Link
Lines compiled: 85087 PASS 2
Warnings: 86 0
Errors: 0 0
-----
Available memory: 1985K
Success
-----
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
```



COMPETENCIA PARTICULAR: Diseña algoritmos y diagramas de flujo para la solución de problemas técnicos y científicos para el área de telecomunicaciones

RAP 1 Elabora el algoritmo de la solución del problema empleando la técnica HIPO .
RAP 2 Elabora diagramas de flujo de los algoritmos correspondientes de acuerdo a simbología estandarizada

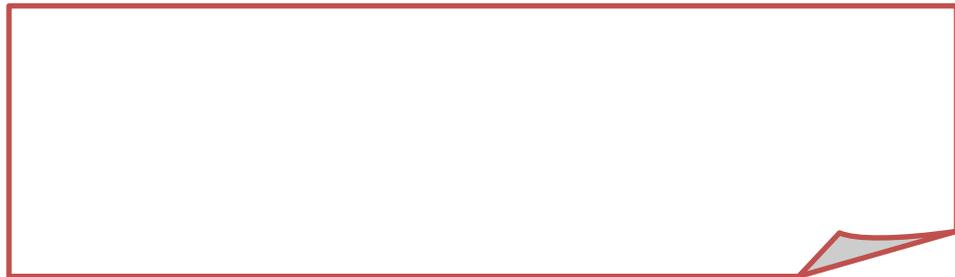
INSTRUCCIONES.

LEE CUIDADOSAMENTE Y ANALIZA LO QUE A CONTINUACIÓN SE TE PIDE PARA REALIZAR LAS SIGUIENTES ACTIVIDADES

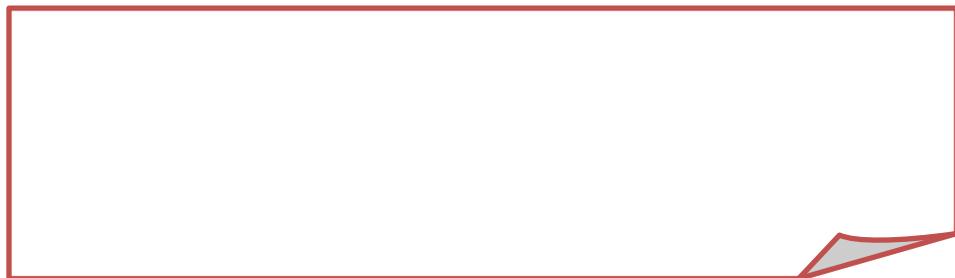
SECCIÓN I.

Algoritmos.

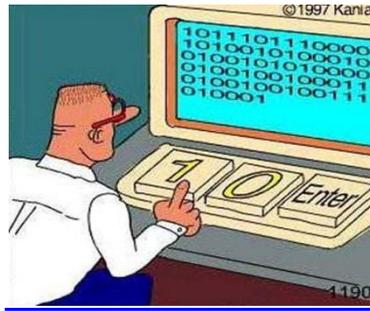
1. Actividad. En una cuartilla escriba el concepto de algoritmo y elabore un mapa conceptual acerca de cómo se desarrollar uno.



2. Actividad. Utilizando la siguiente ecuación $y=3x+10$ elabora en una hoja carta los pasos del algoritmo para que esta pueda ser resuelta correctamente.



3. Actividad. Ilustra en una hoja de papel un procedimiento no matemático (de algún actividad cotidiana) de un algoritmo y sus pasos para resolverlo.

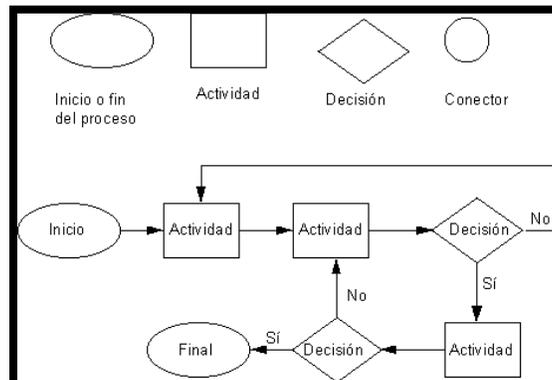


SECCIÓN II.

Diagramas de flujo

4. Actividad. Realice un esquema en donde relacione los principales símbolos de los diagramas de flujo con los procesos relacionados con ellos o su significado.

EJEMPLO :





5. Actividad. De acuerdo a la actividad No 2 de la sección anterior realice el [diagrama de flujo](#) de dicho algoritmo.
6. Actividad. De acuerdo con la actividad No 3 de la sección anterior (La cual debe de ser de una actividad cotidiana ej. Marcación de un número telefónico). Realice un diagrama de flujo de dicho algoritmo.
7. Actividad. Utilizando el diagrama de flujo de la actividad anterior pida a una persona que siga los pasos del diagrama de flujo para completar o solucionar el algoritmo. Cuestiónese si la persona pudo seguir en forma lógica y correcta su procedimiento marcado por el diagrama de flujo para completar la tarea.
8. Actividad. Realice un listado de procedimientos que no funcionan por que las personas no los entienden y que con la ayuda de un diagrama de flujo se podría resolver este problema

UNIDAD 2 DEL PROGRAMA: PROGRAMACIÓN ESTRUCTURADA CON LENGUAJE C

COMPETENCIA PARTICULAR: Codifica las soluciones obtenidas de los diagramas de flujo de acuerdo a las técnicas de la programación estructurada y gramática del lenguaje computacional

RAP 1: Codifica diagramas de flujo de acuerdo a la sintaxis y semántica del lenguaje de programación.

RAP 2: Realiza programas empleando el enfoque de la programación estructurada

INSTRUCCIONES.

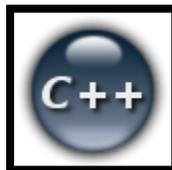


LEE CUIDADOSAMENTE Y ANALIZA LO QUE A CONTINUACIÓN SE TE PIDE PARA REALIZAR LAS SIGUIENTES ACTIVIDADES

SECCIÓN I.

Sintaxis y semántica de las instrucciones básicas del lenguaje C.

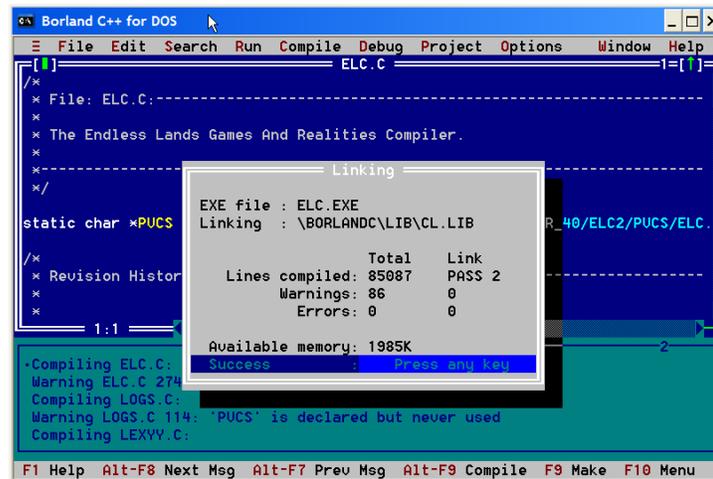
1. Actividad. Elabore una guía practica de las siguientes instrucciones de lenguaje C++, que indique su estructura lógica y una explicación escrita de su funcionamiento ([if else](#) , [switch case](#), [for](#), [while](#)) .
2. Actividad. Relacione cada una de las ordenes expuestas en el punto anterior con su representación en diagramas de flujo.
3. Actividad. Elabore una lista de tipos de variables en C++ clasificándolas.



SECCIÓN II.

Diagramas de flujo

4. Actividad. Realice un Programa en C++ que calcule la ecuación de una recta $y=3x+4$.



5. Actividad. Al programa anterior, agregue una estructura para que en forma automática calcule 100 puntos de la recta utilizando la estructura “for”.
6. Actividad. Elabore un programa en C++, que identifique un número, si este es igual a o menor a 100 y si es mayor indicándolo con un mensaje. Utilice if, else.
7. Actividad. Realice un programa en C++ que realice 5 preguntas numeradas acerca de tu estado de ánimo, formando un menú y que al contestar con el número de pregunta el programa conteste con un consejo diferente para cada pregunta, utilice la orden switch case.
8. Actividad. A todos los anteriores programas incluya una estructura que pregunte al usuario si desea ejecutarlo nuevamente utilizando una variable entera como control y usando la orden “while”, si esta variable es igual a cero continuar si es igual a uno debe de salir del programa.



INSTRUCCIONES: Escriba una instrucción sencilla en C para llevar a cabo las siguientes tareas.

1. Asigne la suma de x y y a z, e incremente al valor de x en 1 después del cálculo.
2. Multiplique la variable producto por 2 mediante el uso del operador *=.



3. Multiplique la variable producto por 2 mediante el uso de los operadores * y =.
4. Verifique si el valor de la variable cuenta es mayor que 10. Si lo es, imprima "cuenta es mayor que 10".
5. Decremento la variable x en 1, después réstela de la variable total.
6. Sume la variable x a la variable total, después decremente x en 1.
7. Calcule el residuo de la división de q entre divisor y asigne el resultado a r. Escriba la instrucción de dos maneras distintas.
8. Imprima el valor 123.4567 con dos dígitos de precisión. ¿Qué valor se imprime?
9. Imprima el valor de punto flotante 3.14159 con tres dígitos de precisión a la derecha del punto decimal. ¿Qué valor se imprime?
10. Inicialice la variable x en 1.

INSTRUCCIONES: Escriba un programa en C que calcule x a la potencia y. El programa debe contener las siguientes instrucciones.

1. Introducir la variable entera x mediante scanf.
2. Introducir la variable entera y mediante scanf.
3. Inicializar la variable entera i en 1.
4. Inicializar la variable entera potencia en 1.
5. Multiplicar la variable potencia por x y asignar el resultado a potencia.
6. Incrementar la variable i en 1.
7. Verificar i para ver se es menor o igual que y en la condición de una instrucción while.
8. Mostrar la variable entera potencia mediante printf.

INSTRUCCIONES: Identifique y corrija los errores en las siguientes instrucciones.

1. while (c <= 5) {
 producto *= c;
 ++c;
2. scanf ("%.4f", %valor);
3. if (genero == 1)
 printf ("Mujer\n");
 else;
 printf ("Hombre\n");



UNIDAD 3 DEL PROGRAMA: PROYECTO

COMPETENCIA PARTICULAR: Desarrolla la solución de proyectos en el área de telecomunicaciones aplicando las técnicas de programación estructurada y paradigmas actuales de computación.

RAP 1: Estructura la solución de un proyecto empleando los arquetipos de la programación estructurada.

RAP 2: Efectúa la codificación y validación del proyecto de acuerdo a las normas del lenguaje computacional y el método de análisis de casos

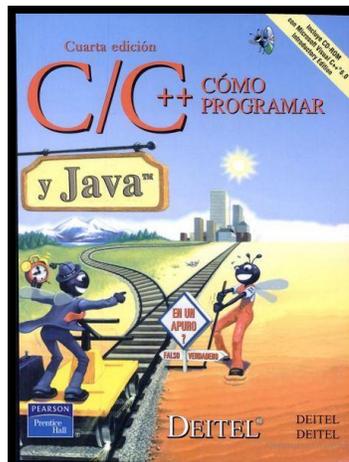
INSTRUCCIONES.

LEE CUIDADOSAMENTE Y ANALIZA LO QUE A CONTINUACIÓN SE TE PIDE PARA REALIZAR LAS SIGUIENTES ACTIVIDADES.

SECCIÓN I.

Solución de proyectos empleando la programación estructurada.

1. Actividad. Elabore un mapa conceptual acerca de la programación estructurada en lenguaje C
2. Actividad. Realice una lista de las ventajas de la programación estructurada.
3. Actividad. Enumere una lista con sus características de los problemas que tiene la programación estructurada.



SECCIÓN II.

Codificación y validación del proyecto mediante el método de análisis de casos.

4. Actividad. [Estructure](#) un programa en lenguaje C el cual utilice arreglos de una sola dimensión para efectuar un promedio de calificaciones.
 5. Actividad. De una serie de números aleatorios desarrolle un programa en c que los ordene de menor a mayor y mayor a menor.
 6. Actividad. Elabore un programa en C++, que contenga por lo menos 5 [procedimientos](#) para sumar, restar, dividir y multiplicar una serie de 20 números aleatorios.
 7. Actividad. Determine el procedimiento para el cálculo de un [determinante matemático](#) y realice un programa para una matriz de 2 x 2.
-



Bibliografía:

Como programar en C/C++ Deitel, H.M.; Deitel P.J. Prentice Hall 2006

Lenguajes de programación C++ Stroustrup, Bjarne Addison Wesley 2006

Fundamentos de programación: Piensa en C Cairó Osvaldo Pearson Educación 2006

Curso de programación con C Ceballos, Francisco Javier Macrobit 2007

Páginas electrónicas

- <http://www.lysator.liu.se/c/>
- www.mis-algoritmos.com
- www.monografias.com



- www.algoritmica.com.ar
- www.chillan.udec.cl