



UNIVERSIDAD SIMÓN BOLÍVAR
Vicerrectorado Académico

1. Departamento: *Computación y Tecnología de la Información (6510)*

2. Asignatura: Laboratorio de Algoritmos y Estructuras I

3. Código de la asignatura: CI2691

No. de unidades-crédito: 2

No. de horas semanales: Teoría 0 Práctica 0 Laboratorio 3

4. Fecha de entrada en vigencia de este programa: Enero 2015

5. OBJETIVO GENERAL: *Al final del curso, el estudiante está capacitado para implementar una solución algorítmica correcta a un problema computacional sencillo, a partir de su especificación formal, haciendo uso de un lenguaje imperativo bajo el enfoque de programación estructurada.*

6. OBJETIVOS ESPECÍFICOS: *El estudiante tendrá competencias para:*

1. *Interpretar un algoritmo especificado en el formalismo GCL*
2. *Especificar las entradas y salidas de un problema usando el lenguaje de la lógica*
3. *Hacer el análisis descendente de un problema computacional*
4. *Concebir una solución algorítmica para un problema computacional sencillo*
5. *Escribir algoritmos usando el formalismo GCL*
6. *Representar información con tipos de datos básicos, ordinales, secuenciales y estructurados no recursivos*
7. *Diseñar programas estructurados que incluyan estructuras de datos y subprogramas*
8. *Procesar datos contenidos en archivos de texto*
9. *Codificar en algún lenguaje imperativo un algoritmo especificado en GCL, siguiendo los estándares de buen estilo de programación*
10. *Verificar las aserciones de un programa y el estado de las variables mediante acciones insertadas en el código como apoyo al proceso de desarrollo*
11. *Redactar informes técnicos sobre la elaboración de programas*
12. *Aplicar prácticas de programación en parejas, ritmo sostenido y desarrollo de casos de prueba, provenientes de los procesos de desarrollo ágil.*

7. CONTENIDOS:

Los laboratorios a ser realizados durante el trimestre son los siguientes:

1. Familiarización con ambiente de trabajo. Estructura general de un programa en un lenguaje imperativo. Declaración de constantes y variables. Tipos Básicos (entero, caracter, real, booleano). Instrucciones de entrada y salida. Especificaciones de entradas y salidas, precondition y postcondición. Comentarios. Documentación del código. (3 Horas)
2. Tipos ordinales. Expresiones aritméticas y lógicas, precedencia. Asignación, instrucción nula y secuenciación. Indentación. Documentación del código. Escritura de aserciones intermedias como expresiones lógicas. (3 Horas)
3. Estructuras de control: selección e iteración. Equivalencias entre instrucciones iterativas. Documentación de código. (3 Horas)
4. Arreglos. Uso de lazos con arreglos. Arreglos Multidimensionales. Iteraciones Anidadas. Constructor de tipos estructurados. Definición de nuevos tipos. Arreglos de estructuras (3 Horas)
5. Verificación de las aserciones y el estado de las variables correspondientes a las precondiciones, postcondiciones y a las instrucciones de asignación, secuenciación y selección, los invariantes y función de cota de una iteración, mediante acciones insertadas en el código. (3 Horas).
6. Análisis descendente. Subprogramas no recursivos. Pasaje de parámetros por valor y por referencia, y su correspondencia con los parámetros de entrada, salida y entrada-salida en GCL. Alcance de identificadores. (3 Horas)
7. Principios de los procesos de desarrollo ágil: prácticas de programación en parejas, ritmo sostenido y desarrollo de casos de prueba (3 Horas)
8. Archivos secuenciales (3 Horas)
9. Recursión (3 Horas)

8. ESTRATEGIAS METODOLÓGICAS, DIDÁCTICAS O DE DESARROLLO DE LA ASIGNATURA:

1. *Talleres cerrados.*
 2. *Trabajos en grupo con un proyecto a resolver fuera del aula.*
 3. *Sesiones de Pre-laboratorio con actividades a resolver fuera del aula antes del taller evaluado.*
- A lo largo del curso se hace énfasis en el buen estilo de programación, programas robustos y prácticas de desarrollo ágil.*

9. ESTRATEGIAS DE EVALUACIÓN:

1. *Se sugiere evaluar la participación en los talleres con un valor de 10%*
2. *Se sugiere realizar varios quices de laboratorio con un valor total de 20%, donde se resuelva un pequeño problema durante una hora; corrigiendo el código atendiendo a criterios de lógica del algoritmo, estilo de programación y robustez.*
3. *Se sugiere realizar un taller evaluado con un valor de 30 % donde se corrija el código atendiendo a criterios de lógica del algoritmo, estilo de programación y robustez, asimismo, se calificará la corrida en base a casos de prueba.*
4. *Se sugiere asignar un proyecto con un valor de 40% dividido en dos entregas de 20%. En cada una debe reportar el trabajo realizado en un informe En la primera entrega se evaluará el análisis descendente de la solución propuesta. En la segunda entrega se evaluará el código (atendiendo a criterios de lógica del algoritmo, estilo de programación y robustez), el funcionamiento del programa y los casos de prueba desarrollados.*

10. FUENTES DE INFORMACIÓN

1. *Manual del lenguaje de programación utilizado*
2. *Oscar Meza. Introducción a la Programación. 2000. Disponible en la web: <http://ldc.usb.ve/~meza/ci-2615/>*
3. *Gries David. "The Science of Programming". Springer-Verlag.1981. ISBN 0-387-96480-0. Páginas 1-85 y 310-319*
4. *Gries David, Schneider Fred. "A Logical Approach to Discrete Math". Springer-Verlag. 1993.*
5. *Jesús Ravelo. Ejemplos de Especificación de Problemas Algorítmicos. 2009. Disponible en la web: <http://ldc.usb.ve/~jravelo/docencia/algoritmos/material/especs.pdf>*

11. CRONOGRAMA DE ACTIVIDADES

Esta sección es un apéndice a ser desarrollado por el profesor al inicio de cada ejecución del programa, y que debe informarse a los estudiantes). Éste orienta al estudiante y al docente sobre el desarrollo de la asignatura en el tiempo. El cronograma puede ser flexible y depende entre otros factores, del período de actividades docentes.

Se presenta el siguiente cronograma como modelo.

Semana	Actividad
1	No hay clase
2	Laboratorio 1
3	Laboratorio 2. Quiz 1
4	Laboratorio 3. Quiz 2.
5	Laboratorio 4. Quiz 3.
6	Laboratorio 5. Quiz 4.
7	TALLER EVALUADO.
8	Laboratorio 6. Enunciado del Proyecto
9	Laboratorio 7.
10	Laboratorio 8. Entrega 1 del proyecto.
11	Laboratorio 9.
12	Entrega 2 del proyecto