Asignatura	**Programación I**
Código	**IS105**
Créditos	**5**
Intensidad semanal	**6 horas semanales para 96 horas totales**
Requisitos	**No tiene**

Justificación	Desde antes de la existencia de los computadores se inventaron métodos para resolver problemas. Los primeros algoritmos conocidos proceden de la antigua Mesopotamia, datan aproximadamente del 3.000 A.C., y estaban escritos en tablillas de arcilla. Su propósito era la realización de cálculos tan pragmáticos como el del capital resultante de un préstamo a interés compuesto y otros similares. Por tanto, no existía la necesidad de hablar en términos de una memoria que cambie por instrucciones en un programa. En la matemática de los últimos cuatrocientos años son muy importantes las funciones. Estas establecen la relación entre los parámetros (la 'entrada') y el resultado (la 'salida') de procesos definidos. Esto es que el resultado depende de una u otra forma de los parámetros. Por esa razón, una función es una buena manera de construir soluciones computacionales y es la base del paradigma de programación funcional. Con el tiempo, al bajar los precios de los computadores y al subir los precios de los programadores, llega a ser más importante describir las soluciones 1 de 11 computacionales en un lenguaje que esté más cerca del 'mundo del hombre', que cerca del computador. Los lenguajes funcionales se unen a la tradición matemática y no están muy influidos por la arquitectura concreta del computador. Para este curso utilizaremos el lenguaje funcional que es un entorno gráfico, interactivo e integrado de programación. **PARADIGMA DECLARATIVO: PROGRAMACIÓN FUNCIONAL** El paradigma declarativo, no se basa en el cómo se hace algo (cómo se logra un objetivo paso a paso), sino que describe (declara) cómo es algo. En otras palabras, se enfoca en describir las propiedades de la solución buscada, dejando indeterminado el algoritmo (conjunto de instrucciones) usado para encontrar esa solución. Esto último se realizará mediante mecanismos internos de inferencia de información a partir de la descripción realizada. La programación funcional es uno de los fundamentales entre los llamados del paradigma declarativo.

	Como tal, permite aunar los componentes de especificación y programación en las tareas de solución automática de problemas. Los lenguajes funcionales ofrecen al programador un buen número de recursos expresivos que permiten resolver problemas complejos mediante programas pequeños y robustos. Entre ellos cabe destacar: 1. Un sistema de tipos polimórficos que permite definir una amplia variedad de estructuras de datos de uso genérico. 2. La posibilidad de definir funciones que aceptan otras funciones como argumentos y devuelven funciones como resultado. 3. Facilidades para definir y manipular estructuras de datos infinitas. 4. Un modelo computacional simple, claro y bien fundamentado, etc. 2 de 1 De no menor importancia es la posibilidad de razonar, de forma sencilla, acerca de las propiedades de los programas: su corrección, su eficacia, su comportamiento en ejecución,.. Esto permite optimizar las tareas de implementación de los lenguajes funcionales. Podemos encontrar, en casi todos los lenguajes de programación funcional, un núcleo común de conceptos y técnicas asentado sobre bases matemáticas firmemente establecidas. En esta asignatura estudiamos dichos y su utilización en la definición de implementaciones correctas y eficientes de los lenguajes de programación que se enmarcan en este paradigma
Objetivo general	Formar en el estudiante las competencias necesarias para formular soluciones a problemas computacionales soportado en el paradigma de programación funcional.
Objetivos Específicos	1. Describir los conceptos principales de la programación funcional. 2. Entender el concepto de tipos de datos. 3. Apropiar y aplicar el concepto de función y recursión 4. Aprender a construir soluciones a partir de funciones.

Metodología	Los estudiantes deberán preparar, antes de la clase, los temas asignados por el profesor. Como apoyo el profesor podrá publicar material en una página Web y/o entregarlo en conferencias. Bajo el esquema de trabajo de esta materia, preparar un tema significa ESTUDIARLO. Estos capítulos pueden ser complementados con la bibliografía que se presenta al final de este documento.

	El trabajo en clase se centrará en presentar los temas en forma magistral, resolver las dudas encontradas por los estudiantes durante la preparación del material, la solución de ejercicios que se hayan asignado, pero sobre todo en discutir nuevos ejercicios que permitan alcanzar mayor claridad en cada tema. También se harán trabajos tendientes a desarrollar en el estudiante la capacidad de traducir a un lenguaje de programación la solución dada a diferentes problemas. Cada grupo podrá contar con un auxiliar de cátedra o monitor que será apoyo, principalmente, en el lenguaje de programación, y quien atenderá a los estudiantes durante la semana en horarios convenidos de común acuerdo con la mayoría de los estudiantes. La asistencia a consultas tanto al profesor como al monitor serán valoradas. Adicionalmente, el profesor atenderá previa cita a los estudiantes, en forma personalizada dos (2) horas adicionales a la semana, en horario que será convenido con los estudiantes. Se recomienda que el profesor fundamente los temas expuestos en clase a través de pruebas de escritorio y planteamiento formal del modelo a resolver.
Competencias Genéricas	Aprendizaje autónomo • Capacidad de análisis y síntesis • Capacidad de aplicar los conocimientos a la práctica • Resolución de problemas • Trabajo individual y por parejas • Comunicación oral y escrita
Competencias específicas	Cognitivas (Saber): • Idioma • Matemáticas • Nuevas tecnologías TIC • Conocimientos de informática • Procedimentales / Instrumentales (Saber hacer): • Redacción en interpretación de documentación técnica • Estimación y programación del trabajo • Planificación, organización y estrategia. • Actitudinales (Ser): • Calidad • Toma de decisión • Capacidad de iniciativa y participación

Estrategias de aprendizaje	Las técnicas docentes que se van a utilizar son: a. Clases de teoría b. Exposiciones sobre trabajos de casos prácticos. c. Desarrollo en clase, de proyectos dirigidos por el docente d. Tutorías colectivas de teoría e. Clases de prácticas f. Corrección de las prácticas g. Tutorías colectivas de prácticas h. Tutorías individualizadas

Contenido de la asignatura	
Unidad 1	**INTRODUCCIÓN A LA INFORMÁTICA.** 1.1 Hardware 1.2 Código binario 1.3 Almacenamiento 1.4 Software: paradigmas, lenguajes de programación, clasificación de software: sistema, desarrollo, aplicativo. 1.5 Estructura de un programa 1.6 Expresiones aritméticas: diferentes notaciones: prefija, infija, sufija, árboles de sintaxis. 1.9 Representación de la información: ASCII, Unicode y BitCode
Unidad 2	**INTRODUCCIÓN AL PARADIGMA FUNCIONAL** 2.1 Características fundamentales de la programación funcional 2.2 Reseña histórica Paradigma Funcional 2.3 Elementos básicos del Paradigma Funcional: 2.3.1 Caracteres, Cadenas de caracteres 2.3.2 Números , Booleanos 2.3.3 Palabras reservadas 2.3.4 Expresiones ,Literales 2.3.5 Operadores aritméticos 2.3.6 Programas simples
Unidad 3	**FUNCIONES** 3.1 Definición de función 3.2 Documentación de código 3.3 Reglas de ámbito léxico 3.4 Definiciones internas 3.5 Composición de funciones
Unidad 4	**PREDICADOS Y SENTENCIAS CONDICIONALES** 4.1 Operadores relacionales 4.2 Operadores lógicos

	4.3 Predicados primitivos 4.4 Predicados simbólicos 4.5 Predicados de equivalentes. 4.6 Formas especiales condicionales: 4.6.1 Forma especial “cond” 4.6.2 Forma especial “cond – else” 4.6.3 Forma especial “if” 4.6.4 Estructuras de bloques
Unidad 5	**RECURSIÓN** 5.1 Recursión simple 5.2 Recursión múltiple 5.3 Manejo de ciclos simples y anidados 5.4 Funciones utilizadas como parámetros
Unidad 6	**TIPOS COMPUESTOS DE DATOS, ORDENAMIENTO Y BÚSQUEDA** 6.1.Estructuras de datos 6.2.Pares 6.3.Listas 6.3.Operaciones con listas
Unidad 7	**GRÁFICOS** 7.1.Introducción a Gráficos 7.2Estructuras y gráficas simples 7.3Operaciones para el manejo del mouse

Semana	***Tema***	***Actividades***	***Referencias Bibliográficas***
1	**INTRODUCCIÓN A LA INFORMÁTICA**	Clase magistral	1
2	**INTRODUCCIÓN AL PARADIGMA FUNCIONAL**	Clase magistral	1,3
3	a. Expresiones aritméticas: b. diferentes notaciones: c. prefija, infija, sufija, d. árboles de sintaxis	Clase magistral talleres, trabajos, quices, consultas	1,4
4	a. Expresiones, Literales b. Variables c. Operadores aritméticos d. Operadores lógico	Clase magistral, talleres, trabajos,	1,2

	e. Operadores relación	quices, consultas	
5	a. Definición de función b. Reglas de ámbito léxico c. Definiciones internas d. Composición de funciones	Clase magistral, talleres, trabajos, quices, consultas	1,3,5
6	Ejercicios y aplicaciones de funciones	Sala de computa Clase magistral, talleres, trabajos, quices, consultas	1,3,4,5
7	a. Predicados primitivos b. Predicados simbólicos c. Predicados numéricos d. Predicados de equivalencia	Clase magistral, talleres, trabajos, quices, consultas	1,2,5
8	a. condicionales b. Formas esp. Condicionales c. Forma especial “if” d. Estructuras de bloques e. Forma especial “cond” f. Forma especial “cond – else”	Clase magistral, talleres, trabajos, quices, consultas	3,4,5
9	a. Manejo de “if “simples y Anidados b. Manejo de “if / else” compuesto y anidados C Manejo “cond” anidado c. Ejercicios	Clase magistral, talleres, trabajos, quices, consultas	3,45
10	a. Funciones b. Funciones utilizadas como parámetros c. Ejercicios de funciones	Sala computo Clase magistral, talleres, trabajos, quices, consultas	1,3,5
11	a. Ejercicios de funciones b. Introducción a Estructuras de datos (concepto)	Sala computo Clase magistral, talleres, trabajos, quices, consultas	1,3,5

12	a. Tipos compuestos de datos b. -Vector c. Operaciones con vectores d. Cadenas. e. Operaciones con cadenas	Sala computo Clase magistral, talleres, trabajos, quinces, consultas	2,4,5
13	a. Ejercicio de Vector b. Ejercicios de Operaciones con vectores	Sala computo Clase magistral, talleres, trabajos, quices, consultas	1,4,5
14	a. Ejercicios de cadenas b. ejercicios con operaciones con cadenas	Sala computo Clase magistral, talleres, trabajos, quices, consultas	1,4,5
15	a. Introducción a Estructuras de datos b. Pares c. Listas d. Operaciones con listas	Sala computo Clase magistral, talleres, trabajos, quices, consultas	1.3,4,5
16	a. Gráficos b. Introducción a Gráficos c. Estructuras y gráficas simples d. Operaciones para el manejo del mouse	Sala computo Clase magistral, talleres, trabajos, quices, consultas	1,2,4,5

Evaluación	***Porcentaje***	***Objetivo***
Parcial 1	***20%***	
Parcial 2	***20%***	

Proyecto	***15%***	
Talleres y laboratorios	***25%***	
Examen final	***20%***	

Texto Guía	MATTHIAS, Felleisen, FINDLER Robert Bruce, FLATT Matthew, KRISHNAMURTHI Shriram, “How to Design Programs An Introduction to Computing and Programming”, The MIT Press, Cambridge, Massachusetts. london, England Last modified: Wednesday, September 24th, 2003 US/Eastern, tomado de internet en: http://www.htdp.org/2003-09-26/Book, el 28 de julio de 2006.

Referencia	***Bibliografía***
1	ABELSON, H., Sussman, G. J. y SUSSMAN, J. “Structure and Interpretations of Computers Programs”. Second edition. The MIT Electrical Engineering and Computers Science Series, 1996. ISBN: 0-262-01153-0.
2	ABELSON, H., Sussman, G. J. y Sussman, J. “Structure and Interpretations of Computers Programs”. The MIT Electrical Engineering and Computers Science Series, 1993. ISBN: 0-262-01077-1.
3	KELSEY, R., CLINGER, W, Rees, J. y otros: “Revised5 Report on the Algorithmic Language Scheme”, 1998.
4	http://www.uco.es/~ma1fegan/manuales/lia/r5rs.pdf HARVEY, B. y WRIGHT, M. “Simply Scheme: Introducing Computer Science”. The MIT Press, 1994. ISBN: 0-262-08226-8.
5	**LINKS DE INTERÉS** http://www.schemers.org. http://www.htdp.org/ http://www.plt-scheme.org/ Grupo de programación funcional del RWTH Aachen (Alemania) http://www-i2.informatik.rwth-aachen.de/Forschung/FP/ Programación funcional en la Universidad Complutense de Madrid http://dalila.sip.ucm.es/funcional/index.html