



Nombre de la asignatura					Programación I				Clave de la asignatura	
Área de formación	Docencia frente a grupo según SATCA				Trabajo de Campo Supervisado según SATCA				Carácter de la asignatura	
	HCS	HPS	TH	C	HTCS	TH	C	TC	(X) Obligatoria	() Optativa
General	1	5	6	6	0	0	0	6		

SERIACIÓN		
Explícita		Implícita
Asignaturas antecedentes	Asignaturas subsecuentes	Conocimientos previos
Algoritmos	Fundamentos de Programación Orientada a Objetos	

PROPÓSITO DE LA ASIGNATURA	
Desarrollar soluciones computacionales utilizando lenguajes de alto nivel para resolver problemas	
COMPETENCIAS A DESARROLLAR	
Genéricas	Específicas
<ol style="list-style-type: none"> 1. Capacidad de análisis y síntesis 2. Resolución de problemas 3. Capacidad de aplicar los conocimientos en la practica 4. Trabajo autónomo 	Desarrollar soluciones integrales, para la mejoría de los procesos en los niveles operativos y tácticos de las organizaciones, facilitando la toma de decisiones aplicando metodologías bajo estándares nacionales e internacionales.



UNIDAD No. 1	Elementos básicos del lenguaje		Horas estimadas para cada unidad
			12
CONTENIDOS			
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje	
1.1. Conceptos generales 1.2. Estructura de un programa 1.3. Datos, tipos de datos y operaciones básicas 1.4. Identificadores, constantes y variables 1.5. Operadores aritméticos, relacionales, lógicos y de asignación. 1.6. Expresiones 1.6.1. Expresiones aritméticas 1.6.2. Reglas de prioridad 1.6.3. Expresiones lógicas 1.7. Entornos de desarrollo integrados para el lenguaje de programación (IDE).	<ul style="list-style-type: none"> • Evalúa expresiones que impliquen el uso combinado de operadores aritméticos lógicos y relacionales. • Codifica en un lenguaje de programación las soluciones a problemas planteados en lenguaje natural y/o fórmulas matemáticas 	<ul style="list-style-type: none"> • Expresiones evaluadas • Código de la solución de un problema 	



UNIDAD No. 2	Estructuras secuenciales		Horas estimadas para cada unidad
			10
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje	
2.1 Instrucciones de entrada y salida 2.2. Funciones predefinidas	<ul style="list-style-type: none"> Resuelve problemas codificando programas en lenguajes de alto nivel empleando estructuras secuenciales 	<ul style="list-style-type: none"> Programas codificados en un lenguaje de programación de alto nivel aplicando estructuras secuenciales. 	



UNIDAD No. 3	Estructuras selectivas	Horas estimadas para cada unidad
		12
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
3.1. Conceptos de estructuras selectivas 3.2. Estructura selectiva simple (if-then) 3.3. Estructura selectiva doble (if-then-else) 3.4. Estructuras selectivas anidadas 3.5. Estructura selectiva múltiple (switch-case)	<ul style="list-style-type: none"> Resuelve problemas codificando programas en lenguajes de alto nivel empleando estructuras selectivas. 	<ul style="list-style-type: none"> Programas codificados en un lenguaje de programación de alto nivel aplicando estructuras selectivas..



UNIDAD No. 4	Estructuras repetitivas	Horas estimadas para cada unidad 12
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
4.1. Conceptos de estructuras repetitivas 4.2. Estructura repetitiva mientras (while) 4.3. Estructura repetitiva hacer-mientras (do-while) 4.4. Estructura desde /para (for)	<ul style="list-style-type: none"> Resuelve problemas codificando programas en lenguajes de alto nivel empleando estructuras repetitivas. 	<ul style="list-style-type: none"> Programas codificados en un lenguaje de programación de alto nivel aplicando estructuras repetitivas..



UNIDAD No. 5	Funciones y modularización		Horas estimadas para cada unidad
			10
CONTENIDOS			
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje	
5.1. Conceptos iniciales sobre funciones y modularización 5.2. Funciones definidas por el programador 5.3. Legibilidad y reusabilidad de código 5.4. Alcance de las variables 5.5. Argumentos por valor y referencia	<ul style="list-style-type: none"> • Crea funciones para ser usadas dentro del código de un lenguaje de programación de alto nivel. • Emplea argumentos por valor y por referencia, en la solución de un problema dentro del código de lenguaje de programación 	<ul style="list-style-type: none"> • Funciones definidas por el programador. • Programas codificados en un lenguaje de programación de alto nivel que utilice funciones definidas por el programador. 	



UNIDAD No. 6	Manejo de cadenas		Horas estimadas para cada unidad
			8
CONTENIDOS			
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje	
6.1. Carácter y cadenas 6.2. Definición de cadenas de caracteres 6.3. Funciones de biblioteca para manejo de cadenas	<ul style="list-style-type: none"> Utiliza las cadenas en la solución de un problema codificado en un lenguaje de programación. 	Programas codificados en un lenguaje de programación de alto nivel que utilice cadenas	



UNIDAD No. 7	Arreglos	Horas estimadas para cada unidad
		16
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
7.1 Introducción a las estructuras de datos 7.2 Arreglos unidimensionales: vectores 7.3 Arreglos mutidimensionales 7.4 Datos de tipo estructura 7.5 Arreglos de estructuras	<ul style="list-style-type: none"> Resuelve problemas codificando programas en lenguajes de alto nivel empleando arreglos. 	<ul style="list-style-type: none"> Programas codificados en un lenguaje de programación de alto nivel que utilice arreglos.



UNIDAD No. 8	Archivos		Horas estimadas para cada unidad
			16
CONTENIDOS			
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje	
8.1 Introducción 8.2 Operaciones primitivas sobre archivos 8.3 Archivos binarios 8.4 Archivos de registros 8.5 Archivos de texto 8.6 Operaciones lógicas sobre archivos	<ul style="list-style-type: none"> Resuelve problemas codificando programas en lenguajes de alto nivel empleando archivos 	<ul style="list-style-type: none"> Programas codificados en un lenguaje de programación de alto nivel que utilice archivos. 	



Contenidos procedimentales	Contenidos actitudinales
<ul style="list-style-type: none"> • Identifica los tipos de datos. • Analiza los requerimientos. • Diseña la solución. • Codifica la solución. • Usa lenguajes de programación. 	<ol style="list-style-type: none"> 1. Participación. 2. Trabajo en equipo. 3. Responsabilidad. 4. Compromiso. 5. Interés. 6. Respeto.

Metodología para la construcción del conocimiento	
Actividades de aprendizaje con el docente	Actividades de aprendizaje autónomo
<ul style="list-style-type: none"> • Prácticas guiadas en el laboratorio de cómputo • Exposición de temas • Prácticas supervisadas 	<ul style="list-style-type: none"> • Desarrollo de programas • Investigación documental y en línea. <ul style="list-style-type: none"> ▪ Uso y análisis de software para el desarrollo de programas. ▪ Reproducción de video tutoriales de lenguajes de programación e entornos para desarrollo integrado • Solución de problemas • Prácticas autónomas

Evidencias de desempeño		
Acreditación	Evaluación	Calificación
<ul style="list-style-type: none"> • La asignatura se acreditará conforme a lo establecido en el Reglamento Escolar vigente. • Entrega de las evidencias de aprendizaje de cada unidad 	Será al final de cada unidad	<ul style="list-style-type: none"> • 5% Expresiones evaluadas. • 5% Código de la solución de un problema. • 10%Programas codificados en un lenguaje de programación de alto nivel aplicando estructuras secuenciales. • 10%Programas codificados en un lenguaje de programación de alto nivel aplicando estructuras selectivas. • 10%Programas codificados en un



		<p>lenguaje de programación de alto nivel aplicando estructuras repetitivas.</p> <ul style="list-style-type: none">• 5%Funciones definidas por el programador.• 10% Programas codificados en un lenguaje de programación de alto nivel que utilice funciones definidas por el programador.• 10%Programas codificados en un lenguaje de programación de alto nivel que utilice cadenas.• 15%Programas codificados en un lenguaje de programación de alto nivel que utilice arreglos.• 20%Programas codificados en un lenguaje de programación de alto nivel que utilice archivos.
--	--	--



BIBLIOGRAFÍA	
BÁSICA	
<ol style="list-style-type: none"> 1. Deitel, P. y Deitel, H. (2010). C Howtoprogram (6a ed.). New Jersey: Pearson Education. 2. Joyanes, L. (2013). Fundamentos generales de programación. México: McGraw-Hill. 3. Joyanes, L. (2012). Fundamentos de la Programación. México: McGraw-Hill. 4. Márquez, G., Osorio, S. y Olvera, N. (2011). Introducción a la programación estructurada en C. México: Pearson educación. 5. Sznajdleder, P. (2012). Algoritmos a fondo con implementaciones en C y Java. México: Alfaomega. 6. Méndez, Al. (2013). Diseño de algoritmos y su programación en C. México: Alfaomega. 	
COMPLEMENTARIAS	
<ol style="list-style-type: none"> 1. Bailey, T. (2005). An introduction to the C programming language and software desing. Recuperado de http://www-personal.acfr.usyd.edu.au/tbailey/ctext/ctext.pdf 2. Gottfried, B. (2005). Programación en C. México: McGraw-Hill. 3. García, F. (2001.). Introducción al lenguaje de programación C. Madrid: Universidad Carlos III, Arcos. 4. Kernighan, B. W., Ritchie, D. M. (2012). The C Programming Language. Massachusetts, EUA: Prentice-Hall 5. Marzal, A. y Gracia, I. (2003). Introducción a la programación con C. Castellón, España: Edición Internet Departamento de Lenguajes y Sistemas informáticos, Universitat Jaume I. Recuperado de http://campuscurico.atalca.cl/~rgarrido/recursos/c.pdf 	

*La bibliografía que se presenta y que tiene una antigüedad mayor a 5 años a la fecha de elaboración de este programa, se justifica debido a que no hay nuevas ediciones y es literatura clásica para la construcción del aprendizaje en esta asignatura.

RESPONSABLE DEL DISEÑO	
Elaborado por	Rubén Jerónimo Yedra Erika Yunuen Morales Mateos
Fecha de elaboración	20 de diciembre de 2016