



Programa de estudio de experiencia educativa

1. Área académica

Área Académica Técnica

2.-Programa educativo

Licenciatura en Matemáticas

3.- Campus

Xalapa

4.-Dependencia/Entidad

Facultad de Matemáticas

5.- Código	6.-Nombre de la experiencia educativa	7.- Área de formación	
		Principal	Secundaria
MTMM 18001	Programación Científica	D	AFEL

8.-Valores de la experiencia educativa

Créditos	Teoría	Práctica	Total horas	Equivalencia (s)
9	3	3	90	Programación I

9.-Modalidad

Curso-Taller

10.-Oportunidades de evaluación

ABGHJK=Todas

11.-Requisitos

Pre-requisitos	Co-requisitos
Ninguno	Ninguno

12.-Características del proceso de enseñanza aprendizaje

Individual / Grupal	Máximo	Mínimo
Grupal	40	10



13.-Agrupación natural de la Experiencia educativa

Academia de métodos matemáticos	Ninguno
---------------------------------	---------

14.-Proyecto integrador

15.-Fecha

Elaboración	Modificación	Aprobación
Enero 2020	---	Junio 2020

16.-Nombre de los académicos que participaron

Dr. Guillermo de Jesús Hoyos Rivera y M.I.A. Patricia González Gaspar

17.-Perfil del docente

Licenciatura en matemáticas, física, físico matemáticas, ingeniero matemático o en áreas relacionadas con ciencias de la computación; con maestría y/o doctorado en ciencias, matemáticas, física o en áreas relacionadas con ciencias de la computación; así como experiencia docente en el área de la computación y experiencia profesional en el ámbito de su disciplina.
--

18.-Espacio

Intraprograma Educativo	Interdisciplinario
-------------------------	--------------------

19.-Relación disciplinaria

20.-Descripción

El curso Programación científica se sitúa en el Área de Formación Disciplinar, con 6 horas a la semana, de las cuales 3 horas son teóricas y 3 horas son prácticas, con un valor de 9 créditos. Éste curso está diseñado para que el estudiante adquiera habilidades para la resolución de problemas de diferentes áreas de las matemáticas, aplicaciones biológicas, financieras, análisis de datos, entre otros; mostrando responsabilidad, dedicación, tolerancia, reflexión y autocrítica. Mediante atención de dudas, asesorías grupales, lluvia de ideas, etc.
--

21.-Justificación

En el quehacer de un matemático se reconocen actividades como la optimización de procesos, predicción de comportamiento futuro de fenómenos, modelado y simulación y análisis de datos; para las cuales, es necesario el uso de programas ad hoc. El avance de la tecnología ha hecho posible el cómputo de métodos matemáticos que resuelven problemas de la matemática misma mediante aproximaciones numéricas, cuando de manera teórica no pueden ser encontradas las soluciones. Por tal motivo es indispensable que el estudiante cuente con la habilidad de abstraer los elementos necesarios del problema para su implementación



computacional. Desde el punto de vista social, existen diferentes problemas que pueden ser estudiados a través de modelos matemáticos que presentan de manera gráfica el comportamiento del mismo; y es a través de simulaciones computacionales del modelo que se proponen alternativas para solucionar los problemas y no exponer a los seres vivos a riesgos innecesarios. Es importante destacar que en el sector privado o público existen instituciones que generan datos masivos derivados de sus actividades y el análisis de dichos datos es sumamente importante para la toma de decisiones.

22.-Unidad de competencia

El estudiante analiza problemas y mediante el diseño de algoritmos identifica los elementos necesarios para el desarrollo de programas computacionales que los resuelvan; ya sea para la implementación de métodos matemáticos, simulación de modelos matemáticos o análisis de datos, mostrando responsabilidad, dedicación, tolerancia, reflexión y autocrítica, con el fin de proponer soluciones a problemas sencillos del área de las Matemáticas.

23.-Articulación de los ejes

Los estudiantes reflexionan en grupo en un marco de orden y respeto mutuo, sobre la importancia del análisis y diseño de algoritmos y programas computacionales; identificando los elementos necesarios para la modelación de soluciones, de manera grupal/equipo mostrarán responsabilidad, dedicación tolerancia y emitirán opiniones con respeto, solidaridad y objetividad.

24.-Saberes

Teóricos	Heurísticos	Axiológicos
<p>Conceptos básicos: Algoritmo, programa, lenguaje de programación, tipos de datos, variables y sentencias.</p> <p>Operadores: Aritméticos, relacionales y lógicos.</p> <p>Estructuras de control: Selectivas (If, If – else, Switch), Iterativas (While, For).</p> <p>Arreglos y matrices</p>	<ul style="list-style-type: none"> • Determinación de fases en la resolución de un problema. • Composición estructural de un programa. • Identificación de tipos de datos básicos. • Modelación de soluciones en pseudocódigo. 	<ul style="list-style-type: none"> • Responsabilidad, dedicación y perseverancia para llevar a cabo las actividades de aprendizaje propuestas para realizar en clase y en casa. • Disposición para someter al escrutinio de los pares el trabajo individual o grupal. • Tolerancia y respeto para aceptar las



<p>Definición de funciones: Parámetro, argumento, valor de retorno.</p> <p>Recursividad</p>	<ul style="list-style-type: none"> • Conocimiento y uso de diferentes operadores • Modificación del flujo de un programa computacional • Rompimiento de secuencialidad • Introducción a las estructuras de datos básicas. • Comprensión de programación modular • Identificación de subprocesos • Discernimiento entre parámetro y argumento • Implementación alternativa de funciones iterativas • Resolución de problemas iterativos 	<p>observaciones y sugerencias.</p> <ul style="list-style-type: none"> • Reflexión autocrítica para mejorar el trabajo Emitir opiniones, identificar y señalar errores e inconsistencias en el trabajo de los compañeros con respeto, solidaridad y objetividad.
---	---	---

25.-Estrategias metodológicas

De aprendizaje	De enseñanza
<ul style="list-style-type: none"> • Lluvia de ideas • Diagramas de flujo • Exposición con apoyo tecnológico variado • Problemario 	<ul style="list-style-type: none"> • Atención a dudas y comentarios • Planteamiento de preguntas guía • Explicación de procedimientos • Asesorías grupales • Supervisión de trabajos

26.-Apoyos educativos

Materiales didácticos	Recursos didácticos
<ul style="list-style-type: none"> • Notas de Clase • Libros 	<ul style="list-style-type: none"> • Proyector de acetatos • Pantalla



<ul style="list-style-type: none"> • Manuales • Documentos de Internet 	<ul style="list-style-type: none"> • Computadora • Cañón de video • Programas computacionales (Compilador del lenguaje de programación elegido para el curso + entorno gráfico de programación, como NetBeans©) • Aula equipada con: gises/plumones, borrador, pintaron/pizarrón, plataforma, mesas duplex, sillas, mesabancos, computadoras
--	--

27.-Evaluación del desempeño

Evidencia (s) de desempeño	Criterios de desempeño	Ámbito(s) de aplicación	Porcentaje
Examen final escrito (Ordinario, Extraordinario y otros establecidos por el Estatuto de los Alumnos)	Resolución acertada de reactivos. Resolución clara y coherente	Aula	100.00%
<p><i>El profesor podrá realizar actividades evaluativas (exámenes parciales, trabajos extra clase, exposiciones, clases prácticas, etc.) durante el periodo escolar y de acuerdo a los resultados de éstas podrá eximir del examen final a aquellos estudiantes que demuestren un alto rendimiento.</i></p>			

28.-Acreditación

Para acreditar esta EE el estudiante deberá haber presentado con idoneidad y pertinencia cada evidencia de desempeño, es decir, que en cada una de ellas haya obtenido cuando menos el 60%, además de cumplir el porcentaje de asistencia establecido en el estatuto de alumnos 2008.

29.-Fuentes de información

Básicas
<ul style="list-style-type: none"> • Bronson G (1999). C++ for engineers and scientists, PWS Publishing. • Harvey M.M. D (2002), C++ how to program, Prentice Hall. • Hubbard, JR (2000). Schaum´s outline of programming with C++, MacGraw-Hill Companies. • Kernighan, BW. & Ritchie, DM (1991). El Lenguaje de Programación C, Segunda Edición, Pearson Education



- Ortega & Grimshaw (1999). An introduction to C++ and numerical analysis, Oxford University Press.
- Steve O (2003), Practical C++ Programming, O'Reilly medi incorporated.

Complementarias

C++ Tutorial - Introduction. [online] Available at:

<https://www.coronadoenterprises.com/tutorials/cpp/cpp_intro.html>

[Acceso Diciembre 6, 2019].

Cómo Aprender Programación (Con Imágenes) - Wikipedi. [online] Available at:

<<https://es.wikihow.com/aprender-programaci%C3%B3n>> [Acceso Diciembre

6, 2019].

MANUAL DE PROGRAMACION BASICA. [online] Available at:

<http://webdelprofesor.ula.ve/ciencias/edbelan/Programacion_Basica/>

[Acceso Diciembre 6, 2019].

Programmer 101: Teach Yourself How To Code. [online] Available at:

<[https://lifelifehacker.com/programmer-101-teach-yourself-how-to-code-](https://lifelifehacker.com/programmer-101-teach-yourself-how-to-code-5401954)

5401954> [Acceso Diciembre 6, 2019].

T5 Matlab 2 | Lenguaje De Programación | Matlab. [online] Available at:

<<https://es.scribd.com/document/440029407/T5-Matlab-2>> [Acceso

Diciembre 6, 2019].

What Programming Language Should I Learn?. [online] Available at:

<[https://regulargeek.com/2009/02/11/what-programming-language-should-i-](https://regulargeek.com/2009/02/11/what-programming-language-should-i-learn/)

learn/> [Acceso Diciembre 6, 2019].