



**UNIVERSIDAD AUTÓNOMA DE ZACATECAS
PLAN ANALÍTICO**

AREA ACADEMICA

Computación

UNIDAD ACADÉMICA	Matemáticas												
PROGRAMA ACADÉMICO	Licenciatura en Matemáticas												
CICLO ESCOLAR	Agosto – Diciembre												
UNIDAD DIDÁCTICA	Introducción a la Programación Funcional				SERIADA CON		Computación Matemática						
EJE CURRICULAR DE LA UNIDAD DIDÁCTICA	Habilidades computacionales												
ACTIVIDAD CON INTERVENCIÓN DOCENTE POR SEMESTRE (Teóricas, Prácticas, a distancia y mixtas)				ACTIVIDAD DE TRABAJO SUPERVISADO POR SEMESTRE				ACTIVIDAD DE TRABAJO INDEPENDIENTE POR SEMESTRE				TOTAL DE HORAS AL SEMESTRE	TOTAL DE CREDITOS DE LA UD
HRS	67.5	CREDITOS	4	HRS	0	CREDITOS	0	HRS	60	CREDITOS	3	127.5	7

COMPETENCIA DE LA UNIDAD DIDÁCTICA

- Construir argumentaciones lógicas, con una clara identificación de hipótesis y conclusiones para sustentar ideas así como proponer o elaborar soluciones a problemas
- Expresar de manera oral y escrita argumentos que sustenten sus razonamientos matemáticos para comunicarse con sus pares e integrantes de equipos de trabajo multidisciplinario
- Implementar definiciones matemáticas mediante un lenguaje de programación la comprensión de conceptos matemáticos involucrados y la solución
- Programar bajo el paradigma funcional un Sistema Algebraico Computacional para la solución de problemas y comprensión del desarrollo de la matemática misma.

UNIDADES DE COMPETENCIA

1. Distinguir entre los paradigmas de programación y las ventajas que ofrece la programación funcional
2. Utilizar el paradigma funcional para realizar programas en un sistema algebraico computacional
3. Manejar las estructuras de datos propias de la programación funcional
4. Crear procedimientos de alto nivel dentro de la programación funcional
5. Resolver problemas de distintas áreas de las matemáticas a través de la programación de un sistema algebraico computacional

MAPA DE ABORDAJE DIDÁCTICO

Unidad 1 – Introducción

Unidad 2 – Cálculo lambda

Unidad 3 – Operaciones sobre listas

Unidad 4 – Procedimientos de alto nivel

Unidad 5 – Programación de conceptos matemáticos

ESCENARIOS

- Aula de Cómputo

ESTRATEGIAS DE ENSEÑANZA

- Lectura de artículos
- Resolución de problemas por medio de la programación
- Exposición de problemas a partir de su origen y resultado (entrada y salida)
- Desarrollo individual de proyectos que involucren la escritura de un reporte y la solución de un problema mediante programación
- Motivación por medio de lecturas y videos de temas matemáticos y computacionales

REQUERIMIENTOS DIDÁCTICOS

LINEAMIENTOS DE EVALUACIÓN Y CERTIFICACIÓN

<ul style="list-style-type: none"> • Computadoras • Bibliografía • Internet • Pizarrón interactivo • Software mathematica 9.0 	<ul style="list-style-type: none"> • Desarrollo de programas que resuelvan las problemáticas planteadas • Elaboración de un proyecto al final del curso que contemple: <ul style="list-style-type: none"> ○ Solución de problemas matemáticos a través de su programación ○ Reporte escrito que describa la motivación, objetivo de su solución, antecedentes históricos y la lógica de programación utilizada. • Presentación • Calidad en trabajos (Limpieza, ortografía, contenido y presentación) • Responsabilidad en el aula (No se permite el uso de celulares, respetar el horario de clases, no interrupciones) • Puntualidad en entrega de trabajos y/o propuestas de solución •
--	--

FUENTES DOCUMENTALES
1.
2.
3.
4.
5.

UNIDAD DE COMPETENCIA 1	TOTAL DE HORAS DEL SEMESTRE QUE SE LLEVA LA UNIDAD DE COMPETENCIA		
	AID	ATS	ATI
Distinguir entre los paradigmas de programación y las ventajas que ofrece la programación funcional			

Desempeños	Saberes Teóricos/Declarativos	Saberes Procedimentales	Competencias Genéricas
<ul style="list-style-type: none"> • Conocer los diferentes paradigmas de programación • Poder evidenciar las ventajas de la programación funcional • Manipular la estructura prefija (polaca-inversa) de las operaciones matemáticas • Conocer los datos y operadores primitivos • Poder construir expresiones compuestas a partir de operadores primitivos • Conocer los procedimientos de primera clase 	<ul style="list-style-type: none"> • Paradigma procedural, orientado a objetos y lógico • Prioridad de operaciones y signos de agrupación • Tipos de datos y operadores • Igualdad entre datos y programas 	<ul style="list-style-type: none"> ○ Estudiar ejemplos de solución de problemas en distintos paradigmas 	<p>Checar</p>

ESTRATEGIA

ESTRATEGIAS DE ENSEÑANZA	ACTIVIDADES DE APRENDIZAJE	
	TRABAJO PRESENCIAL Y/O SUPERVISADO	TRABAJO AUTÓNOMO
<ul style="list-style-type: none"> • Lectura de artículos • Resolución de problemas por medio de la programación • Escritura de un ensayo • Motivación por medio de lecturas y videos de temas matemáticos y computacionales 	<p>Resolver problemas mediante programas</p> <p>Participar en discusiones grupales</p> <p>Manipulación de operaciones que involucren operadores infijos, prefijos y posfijos</p>	<p>Lectura de artículos</p> <p>Escritura de ensayos de las lecturas realizadas</p>

RECURSOS DIDÁCTICOS:

Computadoras
Pizarrón interactivo
Artículos de referencia

EVALUACIÓN

CRITERIOS DE DESEMPEÑO O CALIDAD	EVIDENCIAS	VALOR O PONDERACIÓN
<ul style="list-style-type: none">• Calidad de los Ensayos• Entrega de trabajos propuestos en clase	Trabajos escritos digitales Hojas de registro de actividades	10%

FUENTES DOCUMENTALES

- 1.
- 2.

UNIDAD DE COMPETENCIA 2	TOTAL DE HORAS DEL SEMESTRE QUE SE LLEVA LA UNIDAD DE COMPETENCIA		
	AID	ATS	ATI
Cálculo Lambda Utilizar el paradigma funcional para realizar programas en un sistema algebraico computacional			

Desempeños	Saberes Teóricos/Declarativos	Saberes Procedimentales	Competencias Genéricas
<ul style="list-style-type: none"> Desarrollar programas en un sistema algebraico computacional que soporte procedimientos de primera clase bajo el paradigma funcional Hacer algoritmos recursivos Definir procedimientos anónimos Utilizar instrucciones de control 	<ul style="list-style-type: none"> Definición de algoritmos recursivos Calculo lamba de Church Teoría de estructuras de control Definiciones recursivas de conceptos matemáticos 	<ul style="list-style-type: none"> Programar definiciones matemáticas Programar algoritmos recursivos y estructuras de control Definir funciones anónimas Establecer nombres a funciones y a datos 	Desarrollar programas mediante algoritmos recursivos bajo el paradigma funcional en un sistema algebraico computacional para resolver problemas matemáticos

ESTRATEGIA

ESTRATEGIAS DE ENSEÑANZA	ACTIVIDADES DE APRENDIZAJE	
	TRABAJO PRESENCIAL Y/O SUPERVISADO	TRABAJO AUTÓNOMO
<ul style="list-style-type: none"> Ejemplificar desarrollo de programas a partir de la resolución de un problema Supervisión del desarrollo de programas 	<p>Escribir el algoritmo recursivo del problema planteado</p> <p>Implementar el algoritmo recursivo en el desarrollo de un programa</p>	<p>Escribir el algoritmo recursivo del problema planteado</p> <p>Implementar el algoritmo recursivo en el desarrollo de un programa</p>

RECURSOS DIDÁCTICOS:

Computadoras
Pizarrón
Cañón de proyección
Correo electrónico
Software mathematica

EVALUACIÓN

CRITERIOS DE DESEMPEÑO O CALIDAD	EVIDENCIAS	VALOR O PONDERACIÓN
<ul style="list-style-type: none">• Funcionalidad de los programas• Entrega de algoritmo• Pruebas de escritorio• Ejemplos de la aplicación del programa	<ul style="list-style-type: none">• Trabajos almacenados digitalmente• Registro de participaciones	15%

FUENTES DOCUMENTALES

UNIDAD DE COMPETENCIA 3	TOTAL DE HORAS DEL SEMESTRE QUE SE LLEVA LA UNIDAD DE COMPETENCIA		
	AID	ATS	ATI
Operaciones sobre listas Manejar las estructuras de datos propias de la programación funcional			

Desempeños	Saberes Teóricos/Declarativos	Saberes Procedimentales	Competencias Genéricas
<ul style="list-style-type: none"> Desarrollar programas que manipulan las estructuras de datos en un sistema algebraico computacional bajo el paradigma funcional Hacer algoritmos recursivos sobre la manipulación de listas 	<ul style="list-style-type: none"> Teoría de estructura de datos 	<ul style="list-style-type: none"> Accesar a los elementos de una lista Recorridos sobre listas Manipular listas anidadas en cualquier nivel Creación de listas 	<ul style="list-style-type: none"> Acceder a datos y/o programas que se encuentren almacenados en listas para su procesamiento

ESTRATEGIA

ESTRATEGIAS DE ENSEÑANZA	ACTIVIDADES DE APRENDIZAJE	
	TRABAJO PRESENCIAL Y/O SUPERVISADO	TRABAJO AUTÓNOMO
<ul style="list-style-type: none"> Ejemplificar desarrollo de programas que manipulen listas a partir de la resolución de un problema Supervisión del desarrollo de programas 	<p>Escribir el algoritmo recursivo del acceso a una lista</p> <p>Implementar el algoritmo recursivo en el desarrollo de un programa que acceda a información almacenada en listas</p> <p>Acceder a la información almacenada en cualquier tipo de lista</p>	<p>Escribir el algoritmo recursivo del acceso a una lista</p> <p>Implementar el algoritmo recursivo en el desarrollo de un programa que acceda a información almacenada en listas</p> <p>Acceder a la información almacenada en cualquier tipo de lista</p>

RECURSOS DIDÁCTICOS:

Computadoras
Pizarrón interactivo
Software Mathematica

EVALUACIÓN

CRITERIOS DE DESEMPEÑO O CALIDAD	EVIDENCIAS	VALOR O PONDERACIÓN
<ul style="list-style-type: none">• Funcionalidad de los programas• Entrega de algoritmo• Pruebas de escritorio• Ejemplos de la aplicación del programa	<ul style="list-style-type: none">• Trabajos almacenados digitalmente• Registro de participaciones	15%

FUENTES DOCUMENTALES

UNIDAD DE COMPETENCIA 4 Procedimientos de alto nivel	TOTAL DE HORAS DEL SEMESTRE QUE SE LLEVA LA UNIDAD DE COMPETENCIA		
	AID	ATS	ATI
Crear procedimientos de alto nivel dentro de la programación funcional			

Desempeños	Saberes Teóricos/Declarativos	Saberes Procedimentales	Competencias Genéricas
<ul style="list-style-type: none"> • Desarrollar procedimientos que tengan a procedimiento como argumentos • Manipular programas como datos • Programar procedimientos que dan como resultado procedimientos • Crear estructuras de datos que almacenen procedimientos 	<ul style="list-style-type: none"> • Teoría de procedimientos de primera clase • 	<ul style="list-style-type: none"> • Programar funciones que tienen a procedimientos como argumentos • Programar procedimientos que dan como resultado procedimientos • Realizar programas que creen estructuras de datos que almacenan procedimientos • Desarrollar programas que accedan a procedimientos almacenados 	<ul style="list-style-type: none"> • Crear programas que generan programas para dar soluciones generales a problemas matemáticos.

ESTRATEGIA

ESTRATEGIAS DE ENSEÑANZA	ACTIVIDADES DE APRENDIZAJE	
	TRABAJO PRESENCIAL Y/O SUPERVISADO	TRABAJO AUTÓNOMO
<ul style="list-style-type: none"> • Desarrollar programas que tengan a programas como argumentos • Desarrollar programas que regresen programas como resultado • Desarrollar programas que creen y manipulen estructuras de datos que almacenan procedimientos • Supervisión del desarrollo de programas 	<ul style="list-style-type: none"> • Escribir programas que tengan a programas como argumentos • Escribir programas que regresen programas como resultado • Escribir programas que creen y manipulen estructuras de datos que almacenan procedimientos • Socializar sus propios procedimientos a 	<p>Desarrollar procedimientos que den solución a los problemas planteados</p> <p>Enviar soluciones al profesor</p>

	sus pares mediante exposición de argumentos	
--	---	--

RECURSOS DIDÁCTICOS:

Computadoras
Pizarrón
Software Mathematica

EVALUACIÓN

CRITERIOS DE DESEMPEÑO O CALIDAD	EVIDENCIAS	VALOR O PONDERACIÓN
<ul style="list-style-type: none">Entrega de algoritmoFuncionalidad de los programasPruebas de escritorioEjemplos de la aplicación del programa	<ul style="list-style-type: none">Trabajos almacenados digitalmenteRegistro de participaciones	20%

FUENTES DOCUMENTALES

UNIDAD DE COMPETENCIA 5
Programación de conceptos matemáticos

TOTAL DE HORAS DEL SEMESTRE QUE SE LLEVA LA UNIDAD DE COMPETENCIA

AID	ATS	ATI
------------	------------	------------

Resolver problemas de distintas áreas de las matemáticas a través de la programación de un sistema algebraico computacional

Desempeños	Saberes Teóricos/Declarativos	Saberes Procedimentales	Competencias Genéricas
<ul style="list-style-type: none"> • Desarrollar programas que resuelvan problemas matemáticos específicos de las áreas de: <ul style="list-style-type: none"> ○ Matemática discreta, ○ Álgebra lineal, ○ Estadística, ○ Cálculo diferencial e integral 	<ul style="list-style-type: none"> • Conceptos fundamentales de las áreas de matemática discreta, álgebra lineal, estadística, cálculo diferencial e integral • Elementos de la programación funcional 	<ul style="list-style-type: none"> • Crear programas que resuelvan problemas específicos de las áreas de matemática discreta, álgebra lineal, estadística, cálculo diferencial e integral 	<ul style="list-style-type: none"> • Programar sistemas algebraicos computacionales para automatizar la solución a problemas específicos de las matemáticas.

ESTRATEGIA

ESTRATEGIAS DE ENSEÑANZA	ACTIVIDADES DE APRENDIZAJE	
	TRABAJO PRESENCIAL Y/O SUPERVISADO	TRABAJO AUTÓNOMO
<ul style="list-style-type: none"> • Dar definiciones matemáticas • Plantear problemas específicos de matemáticas • Programar problemas planteados • Supervisar el desarrollo de programas desarrollados por los alumnos 	<ul style="list-style-type: none"> • Escribir programas que resuelvan los problemas planteados 	<p>Desarrollar procedimientos que den solución a los problemas planteados</p> <p>Enviar soluciones al profesor</p>

RECURSOS DIDÁCTICOS:

Computadoras
Pizarrón
Software Mathematica

EVALUACIÓN		
CRITERIOS DE DESEMPEÑO O CALIDAD	EVIDENCIAS	VALOR O PONDERACIÓN
<ul style="list-style-type: none"> • Desarrollo de programas que resuelvan las problemáticas planteadas • Elaboración de un proyecto al final del curso que contemple: <ul style="list-style-type: none"> ○ Solución de problemas matemáticos a través de su programación ○ Reporte escrito que describa la motivación, objetivo de su solución, antecedentes históricos y la lógica de programación utilizada. • Presentación 	<ul style="list-style-type: none"> • Trabajos almacenados digitalmente • CD que contiene el código fuente del programa, reporte y presentación 	40%

FUENTES DOCUMENTALES