



Programa de asignatura por competencias de educación superior

Sección I. Identificación del Curso

Tabla 1. Identificación de la Planificación del Curso.

Actualización:	Marzo 12, 2024		
Carrera:	Ingeniería en Desarrollo de Software	Asignatura:	Optimización
Academia:	Ciencias Computacionales y Programación /	Clave:	19SDSIA03
Módulo formativo:	Informática y Computación	Seriación:	19SDSIA04 - Deep learning
Tipo de curso:	Presencial	Prerrequisito:	19SDSIA02 - Machine learning
Semestre:	Séptimo	Créditos:	6.75
Teoría:	2 horas	Práctica:	2 horas
		Horas semestre:	108 horas
		Trabajo indpt.:	2 horas
		Total x semana:	6 horas

Sección II. Objetivos educacionales

Tabla 2. Objetivos educacionales

Objetivos educacionales		Criterios de desempeño	Indicadores
1	Los egresados gestionarán recursos relacionados con el desarrollo de software en alguna organización.	Los egresados podrán aplicar metodologías en el desarrollo de proyectos en el contexto laboral.	20% de los egresados aplicarán metodologías en el desarrollo de software en su contexto laboral.
2	Los egresados diseñarán e implementarán soluciones innovadoras mediante el uso de tecnologías de la información.	Los egresados participarán activamente en el ciclo de desarrollo e integración continuos	25% de los egresados desempeñarán labores de desarrollo e integración continuos.
3	Los egresados desarrollarán conocimiento especializado que les permite enfocarse en un área del conocimiento específico del desarrollo de software.	Los egresados desempeñarán actividades orientadas al aseguramiento de los activos de información de manera resiliente, la gestión de la infraestructura de redes y comunicaciones, o integrando hardware y software para crear soluciones IoT; así como el uso de inteligencia artificial para gestionar datos y reconocer patrones que determinen oportunidades de negocio en las organizaciones.	5% de los egresados desempeñarán labores en desarrollo de soluciones IoT.
4	Los egresados serán capaces de emprender un negocio basado en el desarrollo de un producto o servicio de tecnologías de la información, aportando valor a la generación de empleos e incrementar el bienestar económico y social, de forma ecológica y sustentable.	Los egresados serán capaces de emprender un negocio basado en el desarrollo propio de un producto o servicio de tecnologías de la información.	2% de los egresados tendrán participación en el acta constitutiva de una empresa creada a partir del desarrollo de software para ofrecer un producto o servicio.



Atributos de egreso de plan de estudios		Criterios de desempeño	Componentes
1	Aplicar y analizar procesos de diseño de ingeniería para generar una experiencia de usuario que asegure cubrir las necesidades como las expectativas de clientes y partes interesadas, utilizando y gestionando la infraestructura de red necesaria.	- Analizará y aplicará procesos de optimización matemáticos como lo son las herramientas de descenso de gradiente para poder encontrar puntos de inflexión en diferentes procesos usados en la industria para la optimización de tiempos y recursos.	1.1 Introducción. 1.2 Métodos clásicos de optimización. 1.2.1 Método del gradiente descendiente. 1.2.2 Cálculo del gradiente. 1.2.3 Ejemplo computacional en MatLab. 1.3 Métodos de cómputo evolutivo. 1.3.1 Procedimiento genérico de un método de cómputo evolutivo. 1.4 Explotación y exploración. 1.5 Aceptación y selección probabilística. 1.5.1 Aceptación y Selección probabilística. 1.5.2 Selección probabilística. 1.6 búsqueda aleatoria. 1.6.1 Ejemplo computacional en MatLab. 1.7 Temple Simulado. 1.7.1 Ejemplo computacional. 2.2.1 Inicialización. 2.2.2 Velocidad de las partículas. 2.2.3 Movimiento de las partículas. 2.2.4 Estructura básica del algoritmo PSO. 2.3 Codificación de PSO. 2.4 Variantes de PSO. 2.4.1 variantes en el proceso de Inicialización. 2.2.2 variantes en la velocidad de las partículas. 2.2.3 Otras mejoras al algoritmo PSO. 2.5 PSO para la optimización con restricciones.



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			2.5.1 Ejemplo.
2	Desarrollar una experimentación adecuada para recopilar, almacenar y analizar grandes cantidades de información basándose en el juicio ingenieril para crear productos o servicios innovadores mediados por software.	- Desarrollará ejercicios de optimización basados en algoritmos bioinspirados. Para conocer la clasificación de diferentes datos, para compararla con el proceso de algún producto o servicio, ver los pro y contras de estas metodologías y compararlas, así como comprender el proceso de procesamiento digital de imágenes	3.1 Introducción. 3.2 Colonia Artificial de Abejas. 3.2.1 Inicialización de la Población. 3.2.2 Enviar a las abejas obreras. 3.2.3 Seleccionar las fuentes de comida por las abejas observadoras. 3.2.4 Determinar a las abejas exploradoras. 3.2.5 Algoritmo Colonia Artificial de Abejas. 3.2.6 Ejemplo computacional MatLab. 3.3 Aplicaciones recientes del algoritmo colonia artificial de abejas en procesamiento de imágenes. 3.3.1 Aplicaciones en el área de procesamiento de imágenes. 3.3.2 Mejoramiento de imagen. 3.3.3 comprensión de imágenes. 3.3.4 detección de bordes. 3.3.5 Clustering. 3.3.6 clasificación de imágenes. 3.3.7 Fusión de imágenes. 3.3.8 análisis de escena. 3.3.9 Reconocimiento de patrones. 3.3.10 Detección de formas. 4.1 Introducción a la optimización inspirada en principios de electromagnetismo. 4.2 optimización inspirada en principios de electromagnetismo. 4.2.1 Inicialización.



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			4.2.2 búsqueda local. 4.2.3 Cálculo del vector de fuerza total. 4.2.4 Movimiento. 4.3 Codificación de EMO.
3	Reconocer la mejora continua como parte de su desarrollo profesional para mantener un perfil actualizado en desarrollo de software para el diseño e implementación de productos y servicios basados en tecnologías con las tendencias emergentes.	- Conocerán otras características propias de la optimización de los algoritmos bioinspirados, heurísticos y metaheurísticos en las diferentes actividades de las matemáticas y física encausados en la industria.	5.1 Introducción a la optimización por búsqueda de armonías. 5.2 Optimización por búsqueda de armonías. 5.2.1 Características de HSA. 5.2.2 Inicialización del problema y de los parámetros de HSA. 5.2.3 Improvisación de los nuevos vectores de armonía. 5.2.4 Actualización de la memoria de armonía. 5.2.5 Configuración de los parámetros HSA. 5.2.6 Procedimiento Computacional. 5.3 Codificación de HSA. 5.4 Variantes de HSA. 5.4.1 Variantes basadas en la configuración de parámetros. 5.4.2 Variantes basadas en la hibridación de HSA con otras técnicas de optimización. 5.4.3 Hibridación de HSA con operadores de otras técnicas metaheurísticas. 5.4.4 hibridación de los componentes HSA en otras técnicas metaheurísticas. 6.1 Introducción. 6.2 Algoritmo de selección Clonal. 6.2.1 Inicialización.



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			6.2.2 Clonación. 6.2.3 Hipermutación. 6.2.4 Reelección. 6.2.5 Introducción de diversidad. 6.3 Pseudocódigo y programa en MatLab. 7.2 Evolución diferencial. 7.2.1 Estructura de la población. 7.2.2 Inicialización.

Sección III. Atributos de la asignatura

Tabla 3. Atributos de la asignatura

Problema a resolver		
Dominar y aplicar las diferentes herramientas de la optimización y resolver problemas específicos que utilizarán las diferentes metodologías en su aplicación.		
Atributos (competencia específica) de la asignatura		
Identificar, formular y resolver problemas complejos de ingeniería aplicando la optimización heurística.		
Aportación a la competencia específica		Aportación a las competencias transversales
Saber	Saber hacer	Saber Ser
<ul style="list-style-type: none"> - Conocer la metodología que ofrece la optimización heurística y metaheurística incluidas las diferentes herramientas como lo son la optimización basado en gradientes o basada en la naturaleza, utilizando MATLAB. - Saber de cierta manera estas metodologías que ayudarán a sacar el mayor rendimiento de estas herramientas en su actuar laboral y /o académico. - Conocer las diferentes herramientas de optimización de recursos y/o servicios, así como posibles predictivos de procesos productivos. - Conocer las diferentes metodologías en la optimización heurística para las posibles aplicaciones de optimización. 	<ul style="list-style-type: none"> - Resolver problemas de aprendizaje utilizando los diferentes modelos para utilizarlos en la optimización de procesos. - Aplicar los conocimientos en la práctica en el desarrollo de la optimización. - Identificar, plantear y resolver problemas específicos acordes a la optimización heurística respaldo y restauración de datos, rendimiento en la optimización y replicación de datos, así como su posible recuperación. 	<ul style="list-style-type: none"> - Entregar los ejercicios propios de optimización en tiempo y forma. - Trabajar en forma autónoma en la configuración de la Optimización. - Reflexión sobre el impacto de las problemáticas en el contexto y su vida cotidiana, así como el aporte de su solución.
Producto integrador de la asignatura, considerando los avances por unidad		
Proyecto integrador, a partir del diseño y/o modificación de una problemática dando solución a la necesidad real de la optimización y toma de decisión, incorporando las competencias desarrolladas en cada una de las unidades de aprendizaje.		

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.1. Desglose específico de la unidad "Optimización y métodos de cómputo evolutivo."

Número y nombre de la unidad: 1. Optimización y métodos de cómputo evolutivo.							
Tiempo y porcentaje para esta unidad:		Teoría:	4 horas	Práctica:	4 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados:		Conocer los antecedentes de lo que hoy conocemos como optimización, así como sus columnas que la originaron, también conocer sus aplicaciones pasadas, actuales y futuras, así como sus aciertos y errores además de lo que esperamos de ella en un futuro cercano.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1.1 Introducción. 1.2 Métodos clásicos de optimización. 1.2.1 Método del gradiente descendiente. 1.2.2 Cálculo del gradiente. 1.2.3 Ejemplo computacional en MatLab. 1.3 Métodos de cómputo evolutivo. 1.3.1 Procedimiento genérico de un método de cómputo evolutivo. 1.4 Explotación y exploración. 1.5 Aceptación y selección probabilística. 1.5.1 Aceptación y Selección probabilística. 1.5.2 Selección probabilística. 1.6 búsqueda aleatoria. 1.6.1 Ejemplo computacional en MatLab. 1.7 Temple Simulado. 1.7.1 Ejemplo computacional.	Saber: - Identificar los orígenes que dan nombre a esta ciencia, así como su repercusión en la humanidad. Saber hacer: - Analizar la relación que tiene con diferentes áreas del conocimiento. Ser: - Entregar los ejercicios propios de optimización en tiempo y forma. - Trabajar en forma autónoma en la configuración de la Optimización.	- Preguntas intercaladas para evaluar los conocimientos previos. - Exposición por parte del profesor de material teórico. - Complementar información con material audiovisual. -Resúmenes. -Mapas conceptuales.	Evaluación diagnóstica - Rescatar conocimientos previos. Evaluación formativa: -Tareas. -Prácticas. Evaluación sumativa: -Examen escrito. -Análisis de caso.	Elaboración de planteamiento y ejercicio en investigación de las columnas que conforman el origen de la optimización, así su distinción de la optimización heurística y metaheurística y su repercusión en otras áreas del conocimiento.			



Continuación: Tabla 4.1. Desglose específico de la unidad "Optimización y métodos de cómputo evolutivo."

Bibliografía

- Cuevas, E.; Rodríguez, A. (2020), Metaheuristic computation with MATLAB®.CRC Press is an imprint of Taylor & Francis Group.
- Cuevas, E.; Osuna, J.V.; Oliva, D.A.; Díaz, M.A. (2016). Optimización. Algoritmos programados con Matlab. México: Alfaomega.

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.2. Desglose específico de la unidad "El algoritmo de optimización por enjambre de partículas (PSO)."

Número y nombre de la unidad: 2. El algoritmo de optimización por enjambre de partículas (PSO).							
Tiempo y porcentaje para esta unidad:		Teoría:	4 horas	Práctica:	4 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados:		Reconocer la diferencia entre las metodologías que implementan algoritmos que utilizan el gradiente descendente como herramienta o los algoritmos heurísticos bioinspirados.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
2.1 Introducción a la optimización de enjambre de partículas. 2.2 Optimización por enjambre de partículas. 2.2.1 Inicialización. 2.2.2 Velocidad de las partículas. 2.2.3 Movimiento de las partículas. 2.2.4 Estructura básica del algoritmo PSO. 2.3 Codificación de PSO. 2.4 Variantes de PSO. 2.4.1 variantes en el proceso de Inicialización. 2.2.2 variantes en la velocidad de las partículas. 2.2.3 Otras mejoras al algoritmo PSO. 2.5 PSO para la optimización con restricciones. 2.5.1 Ejemplo.	Saber: - Identificar las diferentes herramientas la optimización heurística, conceptos base, diseño, función objetivo. Saber hacer: - Resolver ejercicios de procesos de optimización para ponerlos a entrenar con sus respectivos algoritmos. - Identificar los posibles usos de la gradiente descendente. Ser: - Entregar los ejercicios propios de optimización en tiempo y forma.	- Preguntas intercaladas para evaluar los conocimientos previos. - Exposición por parte del profesor de material teórico. - Complementar información con material audiovisual. -Resúmenes. -Mapas conceptuales.	Evaluación formativa: -Tareas. -Prácticas. Evaluación sumativa: -Examen escrito. -Análisis de caso.	Elaboración de planteamiento y ejercicio de un problema laboral o cotidiano en el que se pueda aplicar la optimización en este caso PSO: documentales, conceptos de base, diseño, procesamiento de operaciones.			



Continuación: Tabla 4.2. Desglose específico de la unidad "El algoritmo de optimización por enjambre de partículas (PSO)."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	- Trabajar en forma autónoma en la configuración de la Optimización.			
Bibliografía				
- Cuevas, E.; Rodríguez, A. (2020), Metaheuristic computation with MATLAB®.CRC Press is an imprint of Taylor & Francis Group.				
- Cuevas, E.; Osuna, J.V.; Oliva, D.A.; Diaz, M.A. (2016). Optimización. Algoritmos programados con Matlab. México: Alfaomega.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.3. Desglose específico de la unidad "Algoritmo colonia artificial de abejas (Artificial Bee Colony -ABC)."

Número y nombre de la unidad: 3. Algoritmo colonia artificial de abejas (Artificial Bee Colony -ABC).							
Tiempo y porcentaje para esta unidad:		Teoría:	4 horas	Práctica:	4 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados:		Encontrar las diferencias y las coincidencias entre los algoritmos de optimización bioinspirados que utilizan de cierta manera la inteligencia colectiva para resolver problemas no determinísticos.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
3.1 Introducción. 3.2 Colonia Artificial de Abejas. 3.2.1 Inicialización de la Población. 3.2.2 Enviar a las abejas obreras. 3.2.3 Seleccionar las fuentes de comida por las abejas observadoras. 3.2.4 Determinar a las abejas exploradoras. 3.2.5 Algoritmo Colonia Artificial de Abejas. 3.2.6 Ejemplo computacional MatLab. 3.3 Aplicaciones recientes del algoritmo colonia artificial de abejas en procesamiento de imágenes. 3.3.1 Aplicaciones en el área de procesamiento de imágenes. 3.3.2 Mejoramiento de imagen. 3.3.3 comprensión de imágenes. 3.3.4 detección de bordes. 3.3.5 Clustering.	Saber: - Identificar las diferentes herramientas de la optimización ABC heurística para la optimización de recursos o de procesos. Saber hacer: - Resolver ejercicios de optimización heurística ABC para ponerlos a resolver procesos de optimización. - Identificar los posibles usos de la Optimización ABC. Ser: - Entregar los ejercicios propios de optimización en tiempo y forma.	- Preguntas intercaladas para evaluar los conocimientos previos. - Exposición por parte del profesor de material teórico. - Complementar información con material audiovisual. -Resúmenes. -Mapas conceptuales.	Evaluación formativa: -Tareas. -Prácticas. Evaluación sumativa: -Examen escrito. -Análisis de caso.	Elaboración de planteamiento y ejercicio de un problema industrial de optimización ABC en el que se pueda aplicar la optimización de un proceso o servicio como herramienta práctica para la solución de este.			



Continuación: Tabla 4.3. Desglose específico de la unidad "Algoritmo colonia artificial de abejas (Artificial Bee Colony -ABC)."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
3.3.6 clasificación de imágenes. 3.3.7 Fusión de imágenes. 3.3.8 análisis de escena. 3.3.9 Reconocimiento de patrones. 3.3.10 Detección de formas.	- Trabajar en forma autónoma en la configuración de la Optimización.			
Bibliografía				
- Cuevas, E.; Rodríguez, A. (2020), Metaheuristic computation with MATLAB®.CRC Press is an imprint of Taylor & Francis Group. - Cuevas, E.; Osuna, J.V.; Oliva, D.A.; Diaz, M.A. (2016). Optimización. Algoritmos programados con Matlab. México: Alfaomega.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.4. Desglose específico de la unidad "Algoritmo de optimización inspirado en principios de electromagnetismo (EMO)."

Número y nombre de la unidad: 4. Algoritmo de optimización inspirado en principios de electromagnetismo (EMO).							
Tiempo y porcentaje para esta unidad:		Teoría:	4 horas	Práctica:	4 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados:		Tratar de explicar lo que se entendió por el algoritmo de optimización EMO y tratar de resolver algún problema a través de este algoritmo de optimización.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
4.1 Introducción a la optimización inspirada en principios de electromagnetismo. 4.2 optimización inspirada en principios de electromagnetismo. 4.2.1 Inicialización. 4.2.2 búsqueda local. 4.2.3 Cálculo del vector de fuerza total. 4.2.4 Movimiento. 4.3 Codificación de EMO.	Saber: - Identificar las la optimización inspirada EMO en el electromagnetismo, así como la evaluación de sus diferentes modelos de optimización. Saber hacer: - Resolver ejercicios de optimización EMO, diseño, lenguajes de especificación, consulta, procesamiento de operaciones, de consulta y manejo de transacciones en el contexto de la ingeniería de software.	- Preguntas intercaladas para evaluar los conocimientos previos. - Exposición por parte del profesor de material teórico. - Complementar información con material audiovisual. -Resúmenes. -Mapas conceptuales.	Evaluación formativa: -Tareas. -Prácticas. Evaluación sumativa: -Examen escrito. -Análisis de caso.	Elaboración de planteamiento y ejercicio de un problema laboral o cotidiano en el que se pueda aplicar los diferentes algoritmos de optimización EMO, manejar sus diferentes ramas además de saber utilizarlas adecuadamente.			



Continuación: Tabla 4.4. Desglose específico de la unidad "Algoritmo de optimización inspirado en principios de electromagnetismo (EMO)."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	Ser: - Reflexión sobre el impacto de las problemáticas en el contexto y su vida cotidiana, así como el aporte de su solución.			
Bibliografía				
- Cuevas, E.; Rodríguez, A. (2020), Metaheuristic computation with MATLAB®. CRC Press is an imprint of Taylor & Francis Group. - Cuevas, E.; Osuna, J.V.; Oliva, D.A.; Díaz, M.A. (2016). Optimización. Algoritmos programados con Matlab. México: Alfaomega.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.5. Desglose específico de la unidad "El algoritmo de optimización por búsqueda de armonías (HS)."

Número y nombre de la unidad: 5. El algoritmo de optimización por búsqueda de armonías (HS).							
Tiempo y porcentaje para esta unidad:		Teoría:	5 horas	Práctica:	5 horas	Porcentaje del programa:	13.89%
Aprendizajes esperados:		Tener en cuenta que hay una gama muy extensa en lo que se conoce como la optimización heurística donde se encuentra la optimización bioinspirada y de procesos básicos, pero también existe la optimización metaheurística que es donde se encuentra la optimización HS.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
5.1 Introducción a la optimización por búsqueda de armonías. 5.2 Optimización por búsqueda de armonías. 5.2.1 Características de HSA. 5.2.2 Inicialización del problema y de los parámetros de HSA. 5.2.3 Improvisación de los nuevos vectores de armonía. 5.2.4 Actualización de la memoria de armonía. 5.2.5 Configuración de los parámetros HSA. 5.2.6 Procedimiento Computacional. 5.3 Codificación de HSA. 5.4 Variantes de HSA. 5.4.1 Variantes basadas en la configuración de parámetros.	Saber: - Identificar las diferentes herramientas de la Optimización HS para poder resolver diferentes problemas en la industria y /o academia. Saber hacer: - Resolver ejercicios de optimización para entender los conceptos de base, diseño, procesamiento de operaciones, de consulta y manejo en la optimización HS	- Preguntas intercaladas para evaluar los conocimientos previos. - Exposición por parte del profesor de material teórico. - Complementar información con material audiovisual. -Resúmenes. -Mapas conceptuales.	Evaluación formativa: -Tareas. -Prácticas. Evaluación sumativa: -Examen escrito. -Análisis de caso.	Elaboración de planteamiento y ejercicio de un problema laboral o cotidiano en el que se pueda aplicar las diferentes herramientas de la optimización HS, conceptos base, diseño, lenguajes de especificación, consulta, procesamiento de procesos.			



Continuación: Tabla 4.5. Desglose específico de la unidad "El algoritmo de optimización por búsqueda de armonías (HS)."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
5.4.2 Variantes basadas en la hibridación de HSA con otras técnicas de optimización. 5.4.3 Hibridación de HSA con operadores de otras técnicas metaheurísticas. 5.4.4 hibridación de los componentes HSA en otras técnicas metaheurísticas.	de procesos en el contexto de la ingeniería de software. Ser: - Reflexión sobre el impacto de las problemáticas en el contexto y su vida cotidiana, así como el aporte de su solución.			
Bibliografía				
- Cuevas, E.; Rodríguez, A. (2020), Metaheuristic computation with MATLAB®.CRC Press is an imprint of Taylor & Francis Group. - Cuevas, E.; Osuna, J.V.; Oliva, D.A.; Diaz, M.A. (2016). Optimización. Algoritmos programados con Matlab. México: Alfaomega.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.6. Desglose específico de la unidad "Sistemas inmunes artificiales."

Número y nombre de la unidad: 6. Sistemas inmunes artificiales.							
Tiempo y porcentaje para esta unidad:		Teoría:	5 horas	Práctica:	5 horas	Porcentaje del programa:	13.89%
Aprendizajes esperados:		Analizar este algoritmo que está dentro de los llamados algoritmos de optimización metaheurísticos donde por ejemplo este se inspira o se obtiene del sistema inmune humano, para la solución de problemáticas industriales.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
6.1 Introducción 6.2 Algoritmo de selección Clonal 6.2.1 Inicialización 6.2.2 Clonación 6.2.3 Hipermutación 6.2.4 Reelección 6.2.5 Introducción de diversidad. 6.3 Pseudocódigo y programa en MatLab . 6.3.1 Ejercicios.	Saber: - Identificar las diferentes herramientas del algoritmo de optimización inspirado en el sistema inmune humano para poder resolver diferentes tipos de problemas industriales Saber hacer: - Resolver ejercicios de optimización y tener los conceptos de base, diseño, procesamiento de operaciones, de consulta y manejo del algoritmo de	- Preguntas intercaladas para evaluar los conocimientos previos. - Exposición por parte del profesor de material teórico. - Complementar información con material audiovisual. -Resúmenes. -Mapas conceptuales.	Evaluación formativa: -Tareas. -Prácticas. Evaluación sumativa: -Examen escrito. -Análisis de caso.	Elaboración de planteamiento y ejercicio de un problema laboral o industrial en el que se pueda aplicar las diferentes herramientas del algoritmo del sistema artificial inmune en problemas matemáticos y/o industriales.			



Continuación: Tabla 4.6. Desglose específico de la unidad "Sistemas inmunes artificiales."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	<p>sistema inmune artificial sus respectivas materias.</p> <p>Ser:</p> <ul style="list-style-type: none">- Reflexión sobre el impacto de las problemáticas en el contexto y su vida cotidiana, así como el aporte de su solución.			
Bibliografía				
<ul style="list-style-type: none">- Cuevas, E.; Rodríguez, A. (2020), Metaheuristic computation with MATLAB®.CRC Press is an imprint of Taylor & Francis Group.- Cuevas, E.; Osuna, J.V.; Oliva, D.A.; Diaz, M.A. (2016). Optimización. Algoritmos programados con Matlab. México: Alfaomega.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.7. Desglose específico de la unidad "Algoritmo evolución diferencial (Differential Evolution-DE)."

Número y nombre de la unidad: 7. Algoritmo evolución diferencial (Differential Evolution-DE).							
Tiempo y porcentaje para esta unidad:		Teoría:	5 horas	Práctica:	5 horas	Porcentaje del programa:	13.89%
Aprendizajes esperados:		Encontrar y resolver los diferentes retos que ofrece este algoritmo de optimización llamado de evolución diferencial (Differential Evolution-DE) que se pretenderá utilizar en la resolución de algún problema de optimización.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
7.1 Introducción. 7.2 Evolución diferencial. 7.2.1 Estructura de la población. 7.2.2 Inicialización. 7.2.3 Mutación. 7.2.4 Cruce. 7.2.5 Selección. 7.2.6 Algoritmo evolución diferencial. 7.2.7 Ejemplo computacional en MatLab.	Saber: - Identificar las diferentes herramientas de la evolución diferencial DE, en inspección y seguimiento de modelos de diferentes patrones. Saber hacer: - Resolver ejercicios de optimización para ponerlos a prueba en procesos de optimización DE, en el algoritmo de evolución diferencial. - Tener la experiencia necesaria para poder saber en qué tema en específico a utilizar	- Preguntas intercaladas para evaluar los conocimientos previos. - Exposición por parte del profesor de material teórico. - Complementar información con material audiovisual. -Resúmenes. -Mapas conceptuales.	Evaluación formativa: -Tareas. -Prácticas. Evaluación sumativa: -Examen escrito. -Análisis de caso.	Prácticas que se realizan para resolver diferentes tipos de problemas de optimización DE, de posibles procesos industriales y/o académicos.			



Continuación: Tabla 4.7. Desglose específico de la unidad "Algoritmo evolución diferencial (Differential Evolution-DE)."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	<p>las diferentes herramientas utilizadas en la optimización DE.</p> <p>Ser:</p> <ul style="list-style-type: none"> - Reflexión sobre el impacto de las problemáticas en el contexto y su vida cotidiana, así como el aporte de su solución. 			
Bibliografía				
<ul style="list-style-type: none"> - Cuevas, E.; Rodríguez, A. (2020), Metaheuristic computation with MATLAB®.CRC Press is an imprint of Taylor & Francis Group. - Cuevas, E.; Osuna, J.V.; Oliva, D.A.; Diaz, M.A. (2016). Optimización. Algoritmos programados con Matlab. México: Alfaomega. 				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.8. Desglose específico de la unidad "Algoritmos Genéticos (AG)."

Número y nombre de la unidad: 8. Algoritmos Genéticos (AG).							
Tiempo y porcentaje para esta unidad:		Teoría:	5 horas	Práctica:	5 horas	Porcentaje del programa:	13.89%
Aprendizajes esperados:		Analizar Algoritmos Genéticos (AG) para generarla optimización de secuencia; este algoritmo de optimización es muy poderoso puesto que en la literatura se considera como una escuela aparte de los algoritmos de optimización vistos previamente.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
8.1 Introducción. 8.2 Algoritmos Genéticos. 8.3 Inicialización. 8.3.1 Individuos en formato de números binarios. 8.3.2 Individuos en formato de números reales. 8.4 Selección de padres. 2.4.1 Método de la ruleta. 2.4.2 Sobrante estocástico. 2.4.3 Universal estocástica. 2.4.4 Muestreo determinístico. 2.4.5 Selección de rangos. 2.4.6 Selección por torneo.	Saber: - Identificar las diferentes herramientas de los Algoritmos Genéticos (AG), para la generación de nuevos datos a partir de los ya obtenidos. Saber hacer: - Resolver ejercicios de Algoritmos Genéticos (AG) para ponerlos a entrenar en proceso de clasificación y optimización. - Tener la experiencia necesaria para poder saber en qué tema en específico utilizar	-Preguntas intercaladas para evaluar los conocimientos previos. -Exposición por parte del profesor de material teórico. - Complementar información con material audiovisual. -Resúmenes. -Mapas conceptuales.	Evaluación formativa: -Tareas. -Prácticas. Evaluación sumativa: -Examen escrito. -Análisis de caso.	Solución de un problema de optimización donde se explorará cómo se pueden usar los Algoritmos Genéticos (AG) para generarla optimización de secuencia.			



Continuación: Tabla 4.8. Desglose específico de la unidad "Algoritmos Genéticos (AG)."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
2.4.7 Selección de estado estable. 8.5 Cruza. 8.5.1 Tipos de cruza: individuos binarios e individuos reales. 8.6 Mutación. 8.7 Selección del más apto. 8.8 Pseudocódigo y diagrama de flujo. 8.9 Ejemplo computacional del algoritmo genético en MatLab.	las diferentes herramientas utilizadas en los Algoritmos Genéticos (AG) Ser: - Reflexión sobre el impacto de las problemáticas en el contexto y su vida cotidiana, así como el aporte de su solución.			
Bibliografía				
- Cuevas, E.; Rodríguez, A. (2020), Metaheuristic computation with MATLAB®.CRC Press is an imprint of Taylor & Francis Group. - Cuevas, E.; Osuna, J.V.; Oliva, D.A.; Diaz, M.A. (2016). Optimización. Algoritmos programados con Matlab. México: Alfaomega.				



V. Perfil docente

Tabla 5. Descripción del perfil docente

Perfil deseable docente para impartir la asignatura
<p>Carrera(s): Ingeniero en Sistemas, titulado o carrera a fin o maestría relacionada con el área de conocimiento. o carrera afín</p> <ul style="list-style-type: none">- Con experiencia docente o en el campo deseable de 2 años. Manejo de TIC's. Con habilidades pedagógicas y uso de metodologías alternativas de enseñanza.- Experiencia mínima de dos años- Mínimo Maestría, deseable doctorado.