



Programa de asignatura por competencias de educación superior

Sección I. Identificación del Curso

Tabla 1. Identificación de la Planificación del Curso.

Actualización:	Junio 07, 2022				
Carrera:	Ingeniería en Desarrollo de Software	Asignatura:	Teoría de autómatas		
Academia:	Ciencias Computacionales y Programación /	Clave:	19SDS27		
Módulo formativo:	Programación aplicada	Seriación:	- -		
Tipo de curso:	Presencial	Prerrequisito:	19SDS15 - Estructuras de datos y algoritmia		
Semestre:	Sexto	Créditos:	6.75	Horas semestre:	108 horas
Teoría:	2 horas	Práctica:	2 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 los conocimientos de ciencias básicas como física y matemáticas, así como las ciencias de la ingeniería para generar nuevos productos o servicios basándose en la innovación tecnológica.	- Diseñará aplicaciones eficientes que usen algoritmos especiales de la teoría de autómatas y que cumplan con los principios de diseño de la ingeniería de software.	1. Lenguajes Formales, Alfabetos y Operaciones sobre Lenguajes. 1.1 Introducción a lenguajes formales. 1.2 Introducción a Alfabetos y Cadenas. 1.3 Operaciones Sobre Cadenas. 1.4 Cardinalidad. 1.5 Concatenación. 1.6 Prefijos y Sufijos. 1.7 Operaciones Sobre Lenguajes. 1.8 Cardinalidad. 1.9 Concatenación. 1.10 Unión. 1.11 Intersección. 1.12 Complemento. 1.13 Diferencia. 1.14 Potencia. 1.15 Cerradura de Kleene. 1.16 Cerradura positiva. 1.17 Propiedades de las operaciones sobre lenguajes.
2	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.	- Aplicará algoritmos que puedan interpretar comandos escritos en una fuente de datos, como puede ser un archivo de texto.	2. Gramáticas y Diseño de Gramáticas en Lenguajes Formales. 2.1 Gramáticas. 2.2 Introducción. 2.3 Definición. 2.4 El Lenguaje generado. 2.5 Ambigüedad.



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			2.6 Lenguajes Finitos. 2.7 Lenguajes Infinitos y recurrencia. 2.8 Técnicas de diseño. 2.9 Diseño de Gramáticas. 2.10 Introducción. 2.11 Árboles de Derivación. 2.12 Lenguajes Finitos. 2.13 Recurrencia Estructural. 2.14 Técnica de la Unión. 2.15 Técnica de la Concatenación. 2.16 Técnica de la Cerradura de Kleene. 2.17 Técnica de la Cerradura Positiva.
3	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.	- Diseñará y aplicará algoritmos que se encarguen del análisis de la información vertida en una aplicación o en una fuente de datos.	3. Lenguajes regulares, libres de contexto y Autómatas finitos. 3.1 Lenguajes Regulares. 3.2 Introducción. 3.3 Construcción inductiva. 3.4 Expresiones Regulares. 3.5 Gramáticas Regulares. 3.6 Autómatas Finitos. 3.7 AF Deterministas. 3.8 AF no Deterministas. 3.9 El AFN de una expresión Regular. 3.10 Conversión de un AFN en un AFD. 3.11 Minimización de AFD. 3.12 Lenguajes Libres de contexto. 3.13 Introducción. 3.14 Gramáticas Libres de Contexto.



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			3.15 Implementación de los Autómatas en la resolución de problemas. 3.16 Creación de Manual. 3.17 Herramientas para lenguajes regulares. 3.18 Herramientas para lenguajes no regulares. 3.19 Simplificación mediante división. 3.20 Combinación de las distintas herramientas.

Sección III. Atributos de la asignatura

Tabla 3. Atributos de la asignatura

Problema a resolver		
Analizar, diseñar y codificar algoritmos para automatizar algún proceso de evaluación de datos.		
Atributos (competencia específica) de la asignatura		
Ser capaz de proponer, entender y analizar un algoritmo que resuelva alguna necesidad en cuanto a procesamiento de datos se refiera.		
Aportación a la competencia específica		Aportación a las competencias transversales
Saber	Saber hacer	Saber Ser
<ul style="list-style-type: none"> - Conocer los tipos de datos y algoritmos que se pueden llegar a usar en una aplicación de software. - Conocer las estructuras de datos que se pueden llegar a aplicar en el desarrollo de software. 	<ul style="list-style-type: none"> - Analizar problemas de la vida cotidiana para encontrar un algoritmo que lo represente. - Implementar algoritmos para resolver problemas de índole cotidiana. 	<ul style="list-style-type: none"> - El estudiante trabajará de manera colaborativa con algunos de sus compañeros de clase con lo cual se busca que se desarrolle en él la habilidad de comunicación, responsabilidad y que desarrolle la capacidad de tener una autocrítica y emita su opinión de manera asertiva.
Producto integrador de la asignatura, considerando los avances por unidad		
Desarrollo de un proyecto que se encargue de gestionar los datos vertidos en un archivo, de tal modo que dicho archivo solo debe contener los datos se puedan interpretar como comandos que ejecuten una acción previamente programada.		

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.1. Desglose específico de la unidad "Lenguajes formales, alfabetos y operaciones sobre lenguajes."

Número y nombre de la unidad: 1. Lenguajes formales, alfabetos y operaciones sobre lenguajes.							
Tiempo y porcentaje para esta unidad:		Teoría:	12 horas	Práctica:	12 horas	Porcentaje del programa:	33.33%
Aprendizajes esperados:		- Aplicar diferentes algoritmos sobre cadenas y conjuntos de caracteres con el fin de que una aplicación pueda generar acciones de acuerdo a la interpretación que la aplicación lleve a cabo sobre dichos datos.					
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 a lenguajes formales. 1.2 Introducción a Alfabetos y Cadenas. 1.3 Operaciones Sobre Cadenas. 1.4 Cardinalidad. 1.5 Concatenación. 1.6 Prefijos y Sufijos. 1.7 Operaciones Sobre Lenguajes. 1.8 Cardinalidad. 1.9 Concatenación. 1.10 Unión. 1.11 Intersección. 1.12 Complemento. 1.13 Diferencia. 1.14 Potencia. 1.15 Cerradura de Kleene. 1.16 Cerradura positiva. 1.17 Propiedades de las operaciones sobre lenguajes.	Saber: -Discernir entre los distintos tipos de datos informáticos que pueden intervenir en una aplicación de software. Saber hacer: - Proponer una o varias soluciones a través de algoritmos y pseudocódigos. - Estructurar un programa de acuerdo al algoritmo desarrollado. Ser: -El alumno tendrá la capacidad para trabajar	- Debate sobre los distintos tipos de datos que se usan en la vida cotidiana, lo cual lleve a generar un debate colaborativo. - Analogías de la vida cotidiana en la resolución de problemas y los procesos para resolver problemas computacionalmente. - Usar uno o varios casos de uso para ejemplificar la importancia de los algoritmos en el desarrollo de autómatas y los distintos elementos que intervienen en el desarrollo de un programa.	Evaluación formativa: - Actividades y prácticas. Evaluación formativa: Exámenes, entre escritos y orales. Las evaluaciones orales podrán abordar preguntas específicas de sus trabajos entregados en el portafolio de evidencias.	Portafolio de evidencia: Prácticas de laboratorio y actividades.			



Continuación: Tabla 4.1. Desglose específico de la unidad "Lenguajes formales, alfabetos y operaciones sobre lenguajes."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	de manera colaborativa.	- Cada participante debe generar su portafolio de evidencias, tanto de las notas personales como de los distintos ejercicios de programación.		
Bibliografía				
- Hopcroft, J. E.; Motwani, R.; Ullman, J. D. (2007). Introducción a la teoría de autómatas, lenguajes y computación. Madrid España: Pearson. - Moral, S. (s. f.). Teoría de autómatas y lenguajes formales. Universidad de Granada: Departamento de Ciencias de la Computación e I.A. ETSI.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.2. Desglose específico de la unidad "Gramáticas y diseño de gramáticas en lenguajes formales."

Número y nombre de la unidad: 2. Gramáticas y diseño de gramáticas en lenguajes formales.							
Tiempo y porcentaje para esta unidad:		Teoría:	12 horas	Práctica:	12 horas	Porcentaje del programa:	33.33%
Aprendizajes esperados: - Aplicar los conceptos y algoritmos sobre gramáticas con el fin de generar un analizador sintáctico sobre un conjunto de datos.							
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 Gramáticas. 2.2 Introducción. 2.3 Definición. 2.4 El Lenguaje generado. 2.5 Ambigüedad. 2.6 Lenguajes Finitos. 2.7 Lenguajes Infinitos y recurrencia. 2.8 Técnicas de diseño. 2.9 Diseño de Gramáticas. 2.10 Introducción. 2.11 Árboles de Derivación. 2.12 Lenguajes Finitos. 2.13 Recurrencia Estructural. 2.14 Técnica de la Unión. 2.15 Técnica de la Concatenación. 2.16 Técnica de la Cerradura de Kleene. 2.17 Técnica de la Cerradura Positiva.	Saber: - Conocer los distintos tipos de modelos en la resolución de un problema específico. Saber hacer: - Implementar alguno de los algoritmos para casos particulares de la resolución de problemas en la ingeniería del desarrollo de software. - Usar los distintos tipos de modelos en la resolución de un problema específico. - Discernir sobre cuál es mejor algoritmo para aplicar sobre un problema en	- Cada participante debe generar su portafolio de evidencias, tanto de las notas personales como de los distintos ejercicios de programación. - Plantear un problema o situación en la cual se pueda aplicar la programación y que dé como resultado la solución del problema en cuestión. - Para corroborar los aprendizajes se puede hacer uso de un aula invertida.	Evaluación formativa: - Ejercicios y prácticas. Evaluación sumativa: - Exámenes, entre escritos y orales. Las evaluaciones orales podrán abordar preguntas específicas de sus trabajos entregados en el portafolio de evidencias.	Portafolio de evidencias: Prácticas de laboratorio.			



Continuación: Tabla 4.2. Desglose específico de la unidad "Gramáticas y diseño de gramáticas en lenguajes formales."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	específico del tipo gramatical. Ser: - Trabajo colaborativo. - Habilidad de comunicación asertiva, responsabilidad. - Capacidad de tener una autocrítica y emitir su opinión de manera asertiva.			
Bibliografía				
- Hopcroft, J. E.; Motwani, R.; Ullman, J. D. (2007). Introducción a la teoría de autómatas, lenguajes y computación. Madrid España: Pearson. - Moral, S. (s. f.). Teoría de autómatas y lenguajes formales. Universidad de Granada: Departamento de Ciencias de la Computación e I.A. ETSI.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.3. Desglose específico de la unidad "Estructuras de control en lenguaje C."

Número y nombre de la unidad: 3. Estructuras de control en lenguaje C.							
Tiempo y porcentaje para esta unidad:		Teoría:	12 horas	Práctica:	12 horas	Porcentaje del programa:	33.33%
Aprendizajes esperados:		- Aplicar los conceptos de autómatas finitos en distintos tipos de aplicaciones con el fin de poder analizar estados en un circuito eléctrico o analizar textos grandes y encontrar patrones dentro del mismo texto, entre otros tipos de análisis sobre un conjunto de datos dado.					
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 Lenguajes Regulares. 3.2 Introducción. 3.3 Construcción inductiva. 3.4 Expresiones Regulares. 3.5 Gramáticas Regulares. 3.6 Autómatas Finitos. 3.7 AF Deterministas. 3.8 AF no Deterministas. 3.9 El AFN de una expresión Regular. 3.10 Conversión de un AFN en un AFD. 3.11 Minimización de AFD. 3.12 Lenguajes Libres de contexto. 3.13 Introducción. 3.14 Gramáticas Libres de Contexto.	Saber: - Nombrar los distintos tipos de autómatas que se pueden implementar en el desarrollo de software. Saber hacer: - Proponer de acuerdo a un problema específico, qué tipo autómata de control debería ser utilizado. - Utilizar el mejor modelo de autómata, a su juicio, para resolver alguna parte del problema en general que se esté codificando.	- Cada participante debe generar su portafolio de evidencias, tanto de las notas personales como de los distintos ejercicios de programación que se lleven a cabo como los de tarea. - Plantear un problema o situación en la cual se pueda aplicar la programación y que dé como resultado la solución del problema en cuestión. - Para corroborar los aprendizajes se puede hacer uso de un aula invertida.	Evaluación formativa: - Ejercicios, avance del proyecto. Evaluación sumativa: - Exámenes, entre escritos y orales. Las evaluaciones orales podrán abordar preguntas específicas de sus trabajos entregados en el portafolio de evidencias. - Proyecto final.	Portafolio de evidencias: Proyecto del curso.			



Continuación: Tabla 4.3. Desglose específico de la unidad "Estructuras de control en lenguaje C."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	Ser: - Trabajo colaborativo. - Habilidad de comunicación asertiva, responsabilidad. - Capacidad de tener una autocrítica y emita su opinión de manera asertiva.			
Bibliografía				
- Hopcroft, J. E.; Motwani, R.; Ullman, J. D. (2007). Introducción a la teoría de autómatas, lenguajes y computación. Madrid España: Pearson. - Moral, S. (s. f.). Teoría de autómatas y lenguajes formales. Universidad de Granada: Departamento de Ciencias de la Computación e I.A. ETSI.				



V. Perfil docente

Tabla 5. Descripción del perfil docente

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