

SÍLABO

Programación lógica y funcional (100000S01F)

2025 - Ciclo 2 Agosto

1. DATOS GENERALES

1.1. Carrera:	Ingeniería de Software
1.2. Créditos:	4
1.3. Enseñanza de curso:	Presencial
1.4. Horas semanales:	4

2. FUNDAMENTACIÓN

El curso es relevante para la formación profesional, ya que introduce al estudiante en paradigmas actuales de programación declarativa, los cuales se diferencian de la programación imperativa por emplear expresiones más cercanas al lenguaje natural. Esta perspectiva permite desarrollar soluciones eficientes en tiempo y costo para problemas relacionados con la toma de decisiones basada en el conocimiento, que serían complejos de abordar mediante enfoques imperativos. El curso contribuirá a que el estudiante utilice lenguajes y estructuras declarativas para resolver problemas propios de su campo profesional.

3. SUMILLA

El curso es de naturaleza teórica. Aborda los conceptos básicos de la programación declarativa, incluyendo la programación funcional, la evaluación perezosa y la programación lógica.

4. LOGRO GENERAL DE APRENDIZAJE

Al finalizar el curso, el estudiante aplica los conceptos de programación lógica y programación funcional para plantear y facilitar soluciones alternas a problemas de negocios.

5. UNIDADES Y LOGROS ESPECÍFICOS DE APRENDIZAJE

Unidad de aprendizaje 1: Conceptos básicos..	Semana 1 y 2
Logro específico de aprendizaje: El estudiante identifica los paradigmas y lenguajes de programación representativa declarativa. Reconoce las ventajas de la programación declarativa.	
Temario: <ul style="list-style-type: none"> • La programación declarativa e imperativa. • Evaluación de expresiones y conceptos básicos de la programación declarativa. Componentes de un programa lógico. • Definición de funciones. • Reglas de bifurcación binaria y múltiple. Tipos de datos. 	
Unidad de aprendizaje 2: Programación funcional..	Semana 3,4,5,6 y 7
Logro específico de aprendizaje: El estudiante conoce los elementos de la programación funcional. Aplicar la programación funcional en la resolución de problemas reales. Diseña la base de conocimientos y hace uso de las herramientas alternativas y complementarias de programación.	

Temario: <ul style="list-style-type: none"> • Tipo de datos. • Funciones de control de programas : corte y fallo Ejercicios ilustrativos • Funciones de tipo recursivo. • Formas diversa de implementar funciones recursivas a nivel de la programación lógico. Ejercicios ilustrativos. • Funciones recursivas de tipo menú. De tipo opciones. Intervalos. Operadores. • Conceptos de listas y funtores. • Ejemplo de funtores. Aplicaciones de listas. • Recursividad con listas. • Aplicaciones de recursividad con listas. 	
Unidad de aprendizaje 3: Evaluación perezosa..	Semana 8,9 y 10
Logro específico de aprendizaje: El estudiante identifica la evaluación perezosa como una de las funciones de la programación representativa. Aplica la evaluación perezosa en la resolución de problemas. Diseña programas con modularidad.	
Temario: <ul style="list-style-type: none"> • Listas y cadenas. • Base de Datos dinámicas y externas. • Aplicaciones con Base de datos dinámicas. • Aplicaciones con Base de datos externas. • Estrategia de evaluación perezosa. Técnicas de programación de evaluación perezosa. 	
Unidad de aprendizaje 4: Programación lógica..	Semana 11,12,13,14,15,16,17 y 18
Logro específico de aprendizaje: El estudiante conoce las ventajas y desventajas del paradigma de programación lógica. Aplica la programación lógica en la resolución de problemas reales. Diseña sistemas expertos con programación lógica.	
Temario: <ul style="list-style-type: none"> • Lógica de primer orden. • Unificación y resolución. Cláusulas de Horn y Resolución SLD. • Semántica de los programas lógicos. • Semántica de los programas lógicos. aplicación. • Representación clausada del conocimiento. • Representación clausada del conocimiento. Aplicaciones. • Consulta de una base de cláusulas. • Espacios de búsqueda. • Programación lógica con números, listas y árboles. • Control de búsqueda en programas lógicos. • Control de búsqueda en programas lógicos. Aplicaciones. • Manipulación de términos. Predicados metalógicos. • Manipulación de términos. Predicados metalógicos. Aplicaciones. 	

6. METODOLOGÍA

Para el desarrollo de los aprendizajes del curso, una de las estrategias que se propone es la exposición del docente que proporciona la construcción de los conocimientos a partir de ejemplos y casuísticas que faciliten la comprensión. Asimismo, se promueve la participación activa y permanente del estudiante a través del desarrollo de ejercicios, lecturas, absolución de preguntas, en forma individual y grupal (aprendizaje colaborativo) lo que permite un trabajo metacognitivo, a través de la actividad autónoma del estudiante en el desarrollo de las evaluaciones del curso (aprendizaje autónomo). Por ello es importante que el estudiante asista a las clases, habiendo leído los temas correspondientes a cada sesión. Finalmente, se utilizan otros recursos, como: pizarra, multimedia, videos (aprendizaje para la era digital) y comunicación a través de medios complementarios como correos electrónicos para fomentar una mayor interacción con el estudiante.

7. SISTEMA DE EVALUACIÓN

El cálculo del promedio final se hará de la siguiente manera:

$$(20\%)PC1 + (20\%)PC2 + (30\%)PC3 + (30\%)EXFN$$

Donde:

Tipo	Descripción	Semana	Observación
PC1	PRÁCTICA CALIFICADA 1	5	
PC2	PRÁCTICA CALIFICADA 2	10	
PC3	PRÁCTICA CALIFICADA 3	15	
EXFN	EXAMEN FINAL	18	

Indicaciones sobre Fórmulas de Evaluación:

1. La nota mínima aprobatoria final es de 12.
2. El estudiante que no rinde el examen final puede rendir un único examen de rezagado. La nota obtenida en este examen de rezagado reemplaza al examen final no rendido.
El estudiante rinde el examen de rezagado en la fecha programada por la Universidad, previa presentación de solicitud y pago de los derechos por examen de rezagado dispuesto en el tarifario vigente y publicado en Portal del Estudiante. Los exámenes de rezagados se aplican al final del período lectivo y abarcan todos los temas vistos en la asignatura.
3. En caso un estudiante no rinda una práctica calificada (PC) y, por lo tanto, obtenga NSP, este es reemplazado por la nota obtenida en el examen final. Si también tiene NSP en el examen final, este es reemplazado por la nota obtenida en el examen rezagado. Este reemplazo de nota es automático. No es necesario que el estudiante realice trámite alguno para que proceda el remplazo de la nota. En caso de que el alumno tenga más de una práctica calificada no rendida, solo se reemplaza la práctica calificada de mayor peso.

8. FUENTES DE INFORMACIÓN

Bibliografía Base:

- Juan Moreno. *Programación*. RA-MA Editorial. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=31316>

Bibliografía Complementaria:

- Garassini, Maria Elena - Autor. *Diseño de un software para el aprendizaje de la lengua escrita desde un enfoque comunicativo funcional*. Píxel-Bit. *Revista de Medios y Educación*, (30), 49-60. D - Píxel-Bit. *Revista de Medios y Educación*. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=36380>
- Ceruto Cordovés, Taymi - Autor. *Método para obtener predicados difusos a partir de datos utilizando metaheurísticas*. D - Instituto Superior Politécnico José Antonio Echeverría. CUJAE. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=37412>
- Eduardo Moreno - Autor. *Grafos: fundamentos y algoritmos*. Editorial ebooks Patagonia - J.C. Sáez Editor. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=36884>

9. COMPETENCIAS

Carrera	Competencias específicas
Ingeniería de Software	<ul style="list-style-type: none"> • Desarrollo de Software

10. CRONOGRAMA DE ACTIVIDADES

Unidad de aprendizaje	Semana	Sesión	Tema	Actividades y evaluaciones
Unidad 1 Conceptos básicos.	1	1	La programación declarativa e imperativa.	<ul style="list-style-type: none"> • Prueba de entrada.
		2	Evaluación de expresiones y conceptos básicos de la programación declarativa. Componentes de un programa lógico.	<ul style="list-style-type: none"> • Aprender conceptos básicos de programación declarativa.
		3	Definición de funciones.	<ul style="list-style-type: none"> • Conocer las funciones.

	2	4	Reglas de bifurcación binaria y múltiple. Tipos de datos.	<ul style="list-style-type: none"> Conocer las reglas de bifurcación primaria y múltiple.
Unidad 2 Programación funcional.	3	5	Tipo de datos.	<ul style="list-style-type: none"> Conocer los tipos de datos.
		6	Funciones de control de programas : corte y fallo Ejercicios ilustrativos	<ul style="list-style-type: none"> Elaborar programas con funciones de control
	4	7	Funciones de tipo recursivo.	<ul style="list-style-type: none"> Conocer los tipos de funciones.
		8	Formas diversa de implementar funciones recursivas a nivel de la programación lógico. Ejercicios ilustrativos.	<ul style="list-style-type: none"> Elaborar programas con funciones recursivas.
	5	9	Funciones recursivas de tipo menú. De tipo opciones. Intervalos. Operadores.	<ul style="list-style-type: none"> Elaborar programas con funciones recursivas.
		10	Evaluación	<ul style="list-style-type: none"> PRÁCTICA CALIFICADA 1
	6	11	Conceptos de listas y funtores.	<ul style="list-style-type: none"> Conocer listas y funtores.
		12	Ejemplo de funtores. Aplicaciones de listas.	<ul style="list-style-type: none"> Elaborar programas utilizando listas.
	7	13	Recursividad con listas.	<ul style="list-style-type: none"> Identificar la recursividad.
		14	Aplicaciones de recursividad con listas.	<ul style="list-style-type: none"> Elaborar programas con recursividad con listas.
Unidad 3 Evaluación perezosa.	8	15	Listas y cadenas.	<ul style="list-style-type: none"> Conocer listas y cadenas.
		16	Base de Datos dinámicas y externas.	<ul style="list-style-type: none"> Elaborar programas con base de datos.
	9	17	Aplicaciones con Base de datos dinámicas.	<ul style="list-style-type: none"> Elaborar programas con base de datos dinámicas.
		18	Aplicaciones con Base de datos externas.	<ul style="list-style-type: none"> Elaborar programas con base de datos externas.

	10	19	Estrategia de evaluación perezosa. Técnicas de programación de evaluación perezosa.	<ul style="list-style-type: none"> • Utilizar la técnica de evaluación perezosa.
		20	Evaluación	<ul style="list-style-type: none"> • PRÁCTICA CALIFICADA 2
Unidad 4 Programación lógica.	11	21	Lógica de primer orden.	<ul style="list-style-type: none"> • Entender la lógica de primer orden.
		22	Unificación y resolución. Cláusulas de Horn y Resolución SLD.	<ul style="list-style-type: none"> • Aplicar la cláusula de Horn y resolución LSD.
	12	23	Semántica de los programas lógicos.	<ul style="list-style-type: none"> • Conocer la semántica de los programas lógicos.
		24	Semántica de los programas lógicos. aplicación.	<ul style="list-style-type: none"> • Crear programas utilizando semántica.
	13	25	Representación clausada del conocimiento.	<ul style="list-style-type: none"> • Entender la representación clausada del conocimiento.
		26	Representación clausada del conocimiento. Aplicaciones.	<ul style="list-style-type: none"> • Utilizar la base de cláusulas.
	14	27	Consulta de una base de cláusulas.	<ul style="list-style-type: none"> • Realizar consultas de una base de datos de cláusulas.
		28	Espacios de búsqueda.	<ul style="list-style-type: none"> • Conocer los diferentes espacios de búsqueda.
	15	29	Programación lógica con números, listas y árboles.	<ul style="list-style-type: none"> • Elaborar programas con listas y árboles.
		30	Evaluación	<ul style="list-style-type: none"> • PRÁCTICA CALIFICADA 3
	16	31	Control de búsqueda en programas lógicos.	<ul style="list-style-type: none"> • Conocer el control de búsqueda.
		32	Control de búsqueda en programas lógicos. Aplicaciones.	<ul style="list-style-type: none"> • Elaborar programas utilizando control de búsqueda.
	17	33	Manipulación de términos. Predicados metalógicos.	<ul style="list-style-type: none"> • Entender que es la manipulación de términos.

		34	Manipulación de términos. Predicados metalógicos. Aplicaciones.	<ul style="list-style-type: none"> • Elaborar programas con predicados.
	18	35	Evaluación	<ul style="list-style-type: none"> • Examen Final