

**CIBERTEC**

**VISIÓN:** Ser la institución líder de educación superior técnica en el Perú con alcance a nivel nacional.

**MISIÓN:** Formar profesionales íntegros y competentes brindando una educación superior de alta calidad que contribuya al desarrollo económico y ambiental del país.

**I. INFORMACIÓN GENERAL DEL CURSO**

Carrera(s)	:	Arquitectura de Datos Empresariales con Mención en Big Data Developer
Curso	:	LENGUAJE DE CIENCIA DE DATOS I (4360)
Ciclo	:	Tercero
Período	:	2026-1
Horas semanales <sup>1</sup>	:	6
Créditos	:	3
Requisitos	:	ALGORITMOS Y ESTRUCTURA DE DATOS (4683)

**II. INTRODUCCIÓN**

Lenguaje de Ciencia de Datos I es un curso que pertenece a la línea de programación y desarrollo de soluciones Big Data e inicio de la especialidad algorítmica en las carreras de Arquitectura de Datos Empresariales y Análisis de Datos Empresariales. Brinda un conjunto de conocimientos y habilidades que permiten al alumno hacer uso del lenguaje Python como herramienta para manipular los datos y construir funciones con múltiples enfoques.

El curso es teórico-práctico. En la parte teórica explora los conceptos y técnicas asociadas al lenguaje Python para luego experimentar su implementación en la parte práctica. Se inicia con el estudio del lenguaje Python, sus características, tipos de datos, estructuras de control, el control de flujo programático y la creación de funciones. Luego revisa las librerías y recursos que ofrece Python desde la perspectiva de la programación modular, funcional y orientada a objetos. Continúa con el manejo de la persistencia de los datos a través de archivo de texto, csv y binarios. Finalmente, estudia la implementación de estructuras de datos abstractas como pilas, colas y diferentes tipos de listas.

**III. METODOLOGÍA**

El proceso de enseñanza-aprendizaje se basa en el aprendizaje a partir de la experiencia. Busca motivar al estudiante a través de situaciones cercanas a la realidad y propiciar la reflexión para la resolución de problemas en los que se aplican de forma práctica los conocimientos adquiridos. El aprendizaje del curso se consolida con el desarrollo de un proyecto de investigación asesorado por el docente.

Esta metodología contribuye a que el alumno sea protagonista de su aprendizaje individual y colaborativo mientras que el docente asume un rol de planificador, facilitador y guía, creando escenarios que permiten a los alumnos la adquisición de competencias profesionales.

**IV. LOGRO DEL CURSO**

Al término del curso, el alumno implementa una solución de gestión de datos haciendo uso de las técnicas y herramientas de programación vistas en el curso haciendo uso del lenguaje Python, el manejo de archivos y el acceso a una base de datos.

<sup>1</sup> Las horas semanales cubren los temas hasta la semana 7 según lo establecido en la guía de actividades mediante el desarrollo de clases síncronas, mientras que el resto de las horas se completa con autoestudio, evaluación final y certificación, cumpliendo el total de horas del curso.

## V. RESULTADOS DE APRENDIZAJE

Tipo	Resultado	Aporte
General	<b>RAC 5.- Resuelve situaciones y se orienta a resultados</b> El estudiante es capaz de elaborar propuestas creativas desde su área de conocimiento, tomar decisiones relevantes para el logro de los proyectos en los que participa y actuar para asegurar el cumplimiento del alcance de los mismos de forma eficiente.	Formativo
	<b>RAC 7.- Compromiso con la actualización profesional y la mejora continua</b> El estudiante es capaz aprender de forma autónoma nuevas herramientas, técnicas, modelos, estándares o definiciones en su área de conocimiento que incrementen su competencia, eficiencia y productividad.	Formativo
	<b>RAC 8.- Capacidad de liderazgo y trabajo en equipo</b> El estudiante posee y valora hábitos de trabajo efectivo. Asimismo, evidencia el liderazgo necesario para interactuar con actores de diversos niveles de la empresa y con el entorno en el que esta se desenvuelve logrando resolver situaciones del ámbito laboral con un criterio profesional.	Formativo
Específico	<b>RAC 1.- Desarrolla soluciones Big Data considerando la infraestructura requerida</b> El estudiante es capaz de diseñar, desarrollar, probar, mantener e implementar componentes de una solución Big Data considerando el diseño arquitectónico de la solución, la construcción de los componentes de software necesarios y la infraestructura requerida.	Formativo

## VI. UNIDADES DE APRENDIZAJE

UNIDAD 1. Fundamentos de Python	Duración: 12 horas
Logro de la Unidad de Aprendizaje	
<p>Al término de la unidad, el alumno implementa algoritmos secuenciales en lenguaje Python haciendo uso de los diferentes tipos de datos para la entrada y salida de valores.</p> <p><b>Capacidades</b></p> <ol style="list-style-type: none"> <li>Implementa programas haciendo uso de las principales características del lenguaje Python</li> <li>Implementa programas haciendo uso de los tipos de datos simples y de colección de Python.</li> <li>Implementa programas haciendo uso del control del flujo de los procesos mediante estructuras de selección y repetición</li> </ol>	<p><b>Conocimientos</b></p> <p><b>Temario</b></p> <p><b>1.1. Tema 1: El Lenguaje Python (3 horas)</b></p> <ul style="list-style-type: none"> <li>Características</li> <li>Comentarios <ul style="list-style-type: none"> <li>En línea</li> <li>Multilínea</li> </ul> </li> <li>Cadenas de documentación (docstrings)</li> <li>Variables <ul style="list-style-type: none"> <li>Declaración</li> <li>Tipo de dato dinámico</li> <li>Asignación dinámica de memoria</li> <li>Criterios para definir identificadores</li> </ul> </li> <li>Palabras clave</li> <li>Operadores <ul style="list-style-type: none"> <li>Aritméticos</li> <li>De comparación</li> <li>Lógicos</li> <li>Bit a bit</li> <li>De asignación</li> <li>De identidad</li> <li>De membresía</li> <li>Precedencia de operadores</li> </ul> </li> </ul> <p><b>1.2. Tema 2: Tipos de dato (6 horas)</b></p> <ul style="list-style-type: none"> <li>Tipos de datos numéricos <ul style="list-style-type: none"> <li>Números enteros: int</li> <li>Números reales: float</li> <li>Números complejos: complex</li> </ul> </li> <li>Tipo de dato lógico: bool</li> <li>Tipo de datos cadena: str <ul style="list-style-type: none"> <li>Operaciones comunes con cadenas</li> <li>Indexación de cadenas</li> <li>Subcademas</li> <li>Funciones más utilizadas</li> <li>Formateo de cadenas</li> </ul> </li> </ul>

Capacidades	Conocimientos
<ol style="list-style-type: none"> <li>1. Implementa programas haciendo uso de funciones definidas por el usuario.</li> <li>2. Implementa soluciones haciendo uso de funciones avanzadas.</li> <li>3. Implementa soluciones bajo el paradigma de la Programación orientada a Objetos</li> </ol>	<p>1.2.4. Tipo de dato lista: list      1.2.5. Tipo de dato tupla: tuple      1.2.6. Tipo de dato rango: range      1.2.7. Tipo de dato diccionario: dict</p> <p><b>1.3. Tema 3: Control de flujo (3 horas)</b></p> <p>1.3.1. Estructuras de selección       <ul style="list-style-type: none"> <li>1.3.1.1. Selección simple</li> <li>1.3.1.2. Selección doble</li> <li>1.3.1.3. Selección doble encadenada</li> </ul> </p> <p>1.3.2. Estructuras repetitivas       <ul style="list-style-type: none"> <li>1.3.2.1. Repetitiva while</li> <li>1.3.2.2. Repetitiva for</li> <li>1.3.2.3. Enumeradores</li> </ul> </p> <p>1.3.3. Entrada de datos       <ul style="list-style-type: none"> <li>1.3.3.1. Argumentos de ejecución</li> </ul> </p> <p>1.3.4. Salida de datos</p>

z

UNIDAD 2. Programando en Python	Duración: 15 horas
<b>Logro de la Unidad de Aprendizaje</b>	
Al término de la unidad, el alumno implementa programas desarrollados en Python aplicando diferentes enfoques programáticos para incrementar la eficiencia de los procesos de acuerdo con las necesidades de los casos propuestos.	
Capacidades	Conocimientos
<ol style="list-style-type: none"> <li>4. Implementa programas haciendo uso de funciones definidas por el usuario.</li> <li>5. Implementa soluciones haciendo uso de funciones avanzadas.</li> <li>6. Implementa soluciones bajo el paradigma de la Programación orientada a Objetos</li> </ol>	<p><b>Temario</b></p> <p><b>2.1. Tema 4: Programación modular (6 horas)</b></p> <p>2.1.1. Funciones y módulos incorporados       <ul style="list-style-type: none"> <li>2.1.1.1. Funciones incorporadas</li> <li>2.1.1.2. Biblioteca estándar</li> <li>2.1.1.3. Funciones del módulo math</li> <li>2.1.1.4. Funciones del módulo random</li> <li>2.1.1.5. Funciones del módulo datetime</li> </ul> </p> <p>2.1.2. Funciones definidas por el usuario       <ul style="list-style-type: none"> <li>2.1.2.1. Alcance de variables</li> <li>2.1.2.2. Argumentos definidos e indefinidos</li> <li>2.1.2.3. Retorno de valores</li> <li>2.1.2.4. Módulos</li> <li>2.1.2.5. Manejo de Excepciones</li> </ul> </p> <p><b>2.2. Tema 5: Programación funcional (6 horas)</b></p> <p>2.2.1. Funciones recursivas</p> <p>2.2.2. Funciones decoradores</p> <p>2.2.3. Funciones lambda (anónimas)</p> <p>2.2.4. Funciones avanzadas       <ul style="list-style-type: none"> <li>2.2.4.1. Map</li> <li>2.2.4.2. Filter</li> <li>2.2.4.3. Reduce</li> </ul> </p> <p>2.2.5. Comprensión de colecciones</p> <p>2.2.6. Funciones generadoras</p> <p>2.2.7. Iteradores</p> <p><b>2.3. Tema 6: Programación Orientada a Objetos (3 horas)</b></p> <p>2.3.1. Clases</p> <p>2.3.2. Métodos y atributos</p> <p>2.3.3. Herencia</p> <p>2.3.4. Polimorfismo</p>

<b>UNIDAD 3. Persistencia de datos</b>		<b>Duración: 9 horas</b>
<b>Logro de la Unidad de Aprendizaje</b>		
Al término de la unidad, el alumno implementa una aplicación capaz de persistir los datos archivos txt, csv y binarios así como en una base de datos.		
<b>Capacidades</b>		<b>Conocimientos</b>
1. Implementa procesos de lectura y escritura de archivos usando las librerías de Python 2. Implementa el acceso a una base de datos y manipula los datos con sentencias DML		<b>Temario</b> <b>3.1. Tema 7: Manejo de archivos (3 horas)</b> 3.1.1. Archivos txt 3.1.2. Archivos csv 3.1.3. Archivos binarios <b>3.2. Tema 8: Acceso a base de datos (6 horas)</b> 3.2.1. Conexión 3.2.2. Sentencias DML 3.2.3. Transacciones

<b>UNIDAD 4. TDA - Tipos de Datos Abstractos</b>		<b>Duración: 6 horas</b>
<b>Logro de la Unidad de Aprendizaje</b>		
Al término de la unidad, el alumno identifica y explica el uso y aplicación de los principales Tipos de Datos Abstractos en el desarrollo de algoritmos avanzados.		
<b>Capacidades</b>		<b>Conocimientos</b>
1. Identifica y explica los TDA y su aplicación algorítmica.  <u>Evidencia de Aprendizaje</u>  Evaluación Final (EF): Proyecto Evaluación para Certificación		<b>Temario</b> <b>4.1. Tema 9: TDA - Tipos de Datos Abstractos (6 horas)</b> 4.1.1. Pilas y Colas 4.1.2. Listas enlazadas 4.1.2.1. Simples 4.1.2.2. Circulares 4.1.2.3. Dobles 4.1.2.4. Dobles circulares 4.1.3. Búsqueda y ordenamiento 4.1.4. Árboles y grafos

## VII. EVALUACIÓN

### Fórmula del Curso:

$$\text{Promedio Final} = 20\% \text{ (T1)} + 35\% \text{ (T2)} + 45\% \text{ (EF)}$$

### Dónde:

T1: Evaluación 1

T2: Evaluación 2

EF: Evaluación Final (EF): Proyecto

### Cronograma:

TIPO DE EVALUACIÓN	SESIÓN
T1	03
T2	05
EF	07

Consideraciones:

- El sistema de calificación es vigesimal (Nota máxima: 20).
- La nota mínima aprobatoria es 13.
- Ninguna evaluación es susceptible de eliminación.
- La realización de todos los minicuestionarios disponibles en la plataforma (modalidad virtual) otorga un punto de bonificación en la Evaluación Final.
- En caso de que el curso Sí permita rendir un Examen Sustitutorio que reemplace una de las evaluaciones, la inscripción será comunicada oportunamente.
- Para que el estudiante obtenga el certificado de competencia debe aprobar de manera satisfactoria todos los cursos de su ciclo académico y obtener una nota mínima de 16 en cada evaluación de certificado, el cual se rendirá en la semana 08.

## VIII. BIBLIOGRAFÍA

### Bibliografía

#### Electrónica

- Python Software Foundation (2021) Tutorial de Python. Recuperado de <https://docs.python.org/es/3/tutorial/>
- Python Software Foundation (2021) La Biblioteca Estándar de Python. Recuperado de <https://docs.python.org/es/3.10/library/index.html#library-index>
- Miller B. y Ranum D. *Solución de problemas con algoritmos y estructuras de datos usando Python*. Recuperado de <https://runestone.academy/runestone/static/pythoned/index.html>