

# UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

COORDINACIÓN GENERAL DE FORMACIÓN BÁSICA  
COORDINACIÓN GENERAL DE FORMACIÓN PROFESIONAL Y VINCULACIÓN UNIVERSITARIA  
PROGRAMA DE UNIDAD DE APRENDIZAJE

## I. DATOS DE IDENTIFICACIÓN

- 1. Unidad Académica:** Facultad de Ingeniería, Mexicali; Facultad de Ingeniería, Arquitectura y Diseño, Ensenada y Facultad de Ciencias Químicas e Ingeniería, Tijuana.
- 2. Programa Educativo:** Ingeniero en Computación
- 3. Plan de Estudios:** 2020-1
- 4. Nombre de la Unidad de Aprendizaje:** Lenguaje de Programación Python
- 5. Clave:** 36305
- 6. HC:** 02 **HL:** 00 **HT:** 02 **HPC:** 00 **HCL:** 00 **HE:** 02 **CR:** 06
- 7. Etapa de Formación a la que Pertenece:** Básica
- 8. Carácter de la Unidad de Aprendizaje:** Optativa
- 9. Requisitos para Cursar la Unidad de Aprendizaje:** Ninguno



### Equipo de diseño de PUA

Luz Evelia López Chico  
Carelia Guadalupe Gaxiola Pacheco  
Tania Angélica López Chico  
Sergio Omar Infante Prieto

### Vo.Bo. de Subdirectores de Unidades Académicas

Alejandro Mungaray Moctezuma  
Humberto Cervantes de Ávila  
Rocío Alejandra Chávez Santoscoy

**Fecha:** 17 de octubre de 2019

## **II. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE**

Esta unidad de aprendizaje proporciona la habilidad en el uso de una herramienta de programación útil en la solución de problemas de ciencia básica e ingeniería. Es un lenguaje de programación libre, de código abierto, multiplataforma que se utiliza para desarrollar aplicaciones y servicios, y está presente en ámbitos de la industria e investigación. Se imparte en la etapa básica con carácter optativo y pertenece al área de conocimiento Ciencias de la Ingeniería.

## **III. COMPETENCIA DE LA UNIDAD DE APRENDIZAJE**

Desarrollar aplicaciones computacionales en un lenguaje interpretado, para solucionar problemas en los ámbitos de la industria e investigación, tomando en cuenta las ventajas de simplicidad, versatilidad y rapidez de desarrollo de la herramienta tecnológica, con una actitud creativa y propositiva.

## **IV. EVIDENCIA(S) DE DESEMPEÑO**

Elabora y entrega portafolio de evidencias en formato electrónico con las soluciones a problemas resueltos mediante el desarrollo de aplicaciones computacionales; se deberá incluir por problema una reflexión sobre la estrategia de solución.

## V. DESARROLLO POR UNIDADES

### UNIDAD I. Lenguaje de programación interpretado

**Competencia:**

Identificar objetos y estructuras de datos en lenguaje de programación interpretado, a través de la revisión de la definición, estructura y uso de las palabras reservadas, para redactar líneas de código que cumplan con la sintaxis requerida por el lenguaje, de manera paciente y persistente.

**Contenido:****Duración:** 4 horas

- 1.1 Concepto de lenguaje de programación interpretado.
- 1.2 Objetos y estructuras de datos.
  - 1.2.1 Tipos de datos.
  - 1.2.2 Números.
  - 1.2.3 Asignación de variables.
  - 1.2.4 Cadenas.
  - 1.2.5 Operaciones con cadenas.
    - 1.2.5.1 Propiedades de las cadenas y métodos
  - 1.2.6 Listas.
  - 1.2.7 Diccionarios.
  - 1.2.8 Tuplas.
  - 1.2.9 Conjuntos.
  - 1.2.10 Booleanos.

## UNIDAD II. Expresiones declarativas

### **Competencia:**

Integrar en las declaraciones del lenguaje interpretado operadores de comparación, siguiendo las normas establecidas de su uso, para redactar bloques de código que controlen el flujo de las acciones de acuerdo con los requerimientos, de manera clara y organizada.

### **Contenido:**

**Duración:** 12 horas

- 2.1 Operadores de comparación
  - 2.1.1 Operadores lógicos.
- 2.2 Declaraciones
  - 2.2.1 IF, ELIF y ELSE.
  - 2.2.2 FOR
  - 2.3.3 WHILE
  - 2.3.4 Listas de Comprensión.

## UNIDAD III. Métodos y funciones

### **Competencia:**

Utilizar métodos y funciones, siguiendo las normas establecidas en la sintaxis del lenguaje interpretado, para convertir los enunciados que describen los problemas a resolver en código óptimo y funcional, de manera ordenada y creativa.

### **Contenido:**

**Duración:** 16 horas

#### 3.1 Métodos.

##### 3.1.1 Objetos

#### 3.2 Funciones.

##### 3.2.1 Bloque de instrucciones.

##### 3.2.2 Estructura de la función.

###### 3.2.2.1 Argumentos de entrada.

###### 3.2.2.1 Retorno de argumentos.

##### 3.2.3 Llamado de funciones.

#### 3.3 Declaraciones anidadas y alcance.

## VI. ESTRUCTURA DE LAS PRÁCTICAS DE LABORATORIO

No. de Práctica	Competencia	Descripción	Material de Apoyo	Duración
UNIDAD I				
1	Identificar el ambiente de desarrollo del lenguaje, a través del uso de un editor de texto, para escribir líneas de código que cumplan con el formato requerido por el lenguaje, de manera ordenada y sistemática.	<p>El docente proporciona la versión de lenguaje a utilizar, el procedimiento de instalación, opciones del editor a utilizar y las instrucciones para utilizar el ambiente de desarrollo.</p> <p>El alumno:</p> <ol style="list-style-type: none"> <li>1. Identifica la versión del lenguaje y sus características.</li> <li>2. Da seguimiento al procedimiento de instalación.</li> <li>3. Identifica los componentes del ambiente de desarrollo.</li> <li>4. Selecciona un editor de texto de acuerdo al sistema operativo instalado.</li> <li>5. Identifica el procedimiento de uso.</li> <li>6. Realiza las actividades propuestas.</li> <li>7. Verifica que los archivos generados cumplan con los requisitos del ambiente de desarrollo.</li> </ol> <p>Entrega un reporte de las actividades realizadas en formato electrónico.</p>	Pintarrón, Equipo Audiovisual, Equipo de cómputo, documentación del lenguaje, archivos de ayuda audiovisual.	1 hora
2	Distinguir los objetos y estructuras de datos, mediante el uso de la	El docente describe y muestra los tipos de datos básicos,	Pintarrón, Equipo Audiovisual, Equipo de	1 hora

	<p>experimentación, para redactar líneas de código que cumplan con la sintaxis requerida por el lenguaje de programación, de manera paciente y persistente.</p>	<p>declaración de variables, operaciones, propiedades y métodos; proporciona las actividades a realizar. El alumno:</p> <ol style="list-style-type: none"> <li>1. Realiza actividades con números enteros y de punto flotante, utilizando un entorno interactivo de ejecución de código.</li> <li>2. Prueba declaración de cadenas, concatenación, trucado, demostraciones de propiedades de las cadenas y métodos utilizando un entorno interactivo de ejecución de código.</li> <li>3. Realiza impresiones de cadenas utilizando diversos formatos e interpolaciones de cadenas con diferentes tipos de datos utilizando un entorno interactivo de ejecución de código.</li> </ol> <p>Responde cuestionario en línea del tema tipo de datos.</p>	<p>cómputo, documentación del lenguaje, archivos de ayuda audiovisual.</p>	
3		<p>El docente describe y muestra, el tema de listas, diccionarios, tuplas, sets y booleans; proporciona actividades a realizar. El alumno:</p> <ol style="list-style-type: none"> <li>1. Realiza declaraciones y operaciones con listas y diccionarios que contienen diferentes tipos de datos</li> </ol>	<p>Pintarrón, Equipo Audiovisual, Equipo de cómputo, documentación del lenguaje, archivos de ayuda audiovisual.</p>	1 hora

		<p>utilizando un entorno interactivo de ejecución de código.</p> <p>2. Realiza declaraciones y operaciones con tuplas y sets, compara tuplas y listas utilizando un entorno interactivo de ejecución de código.</p> <p>3. Realiza ejercicios demostrativos del funcionamiento de los datos tipo boolean utilizando un entorno interactivo de ejecución de código.</p> <p>Responde cuestionario en línea de listas, diccionarios, tuplas, sets y boolean.</p>		
4		<p>El docente describe, ejemplifica la lectura y escritura en archivos de texto, y las buenas prácticas recomendadas; proporciona actividades a realizar.</p> <p>El alumno:</p> <p>1. Realiza operaciones de lectura, escritura y sobreescritura de archivos de texto, utilizando un entorno interactivo de ejecución de código.</p> <p>Responde cuestionario en línea de lectura, escritura y sobreescritura en archivos de texto.</p>	<p>Pintarrón, Equipo Audiovisual, Equipo de cómputo, documentación del lenguaje, archivos de ayuda audiovisual.</p>	1 hora
<b>UNIDAD II</b>				
5	Utilizar los operadores de	El docente describe, ejemplifica la	Pintarrón, Equipo	2 horas

	<p>comparación, a través de la observación de casos y la experimentación, para escribir sentencias de código que efectúen operaciones de comparación entre valores de variables y entreguen un resultado de tipo booleano, de manera ordenada y determinada.</p>	<p>operación de los operadores de comparación y su concatenación; proporciona actividades a realizar.</p> <p>El alumno:</p> <ol style="list-style-type: none"> <li>1. Realiza operaciones con operadores de comparación y utiliza operadores lógicos para concatenarlos, utilizando un entorno interactivo de ejecución de código.</li> </ol> <p>Responde cuestionario en línea de operadores de comparación.</p>	<p>Audiovisual, Equipo de cómputo, documentación del lenguaje, archivos de ayuda audiovisual.</p>	
6	<p>Utilizar las declaraciones del lenguaje interpretado, a través del análisis de sus conceptos, observación de ejemplos y la experimentación en un entorno interactivo de ejecución, para escribir sentencias de código que efectúen control de flujo o manejo de bifurcaciones y resulten en la ejecución de operaciones específicas, de manera creativa y ordenada.</p>	<p>El docente ejemplifica la forma de utilizar las declaraciones if, elif, else; proporciona actividades a realizar.</p> <p>El alumno:</p> <ol style="list-style-type: none"> <li>1. Realiza una lectura con los conceptos y estructuras del uso de las palabras clave o reservadas if, elif, else.</li> <li>2. Observa un video instruccional para reforzar las lecturas del tema.</li> <li>3. Realiza operaciones con operadores de comparación y utiliza operadores lógicos para concatenarlos, utilizando un entorno interactivo de ejecución de código.</li> </ol> <p>Responde cuestionario en línea del tema de declaraciones con if, elif y else.</p>	<p>Pintarrón, Equipo Audiovisual, Equipo de cómputo, documentación del lenguaje, archivos de ayuda audiovisual.</p>	2 horas

7	<p>Emplear las palabras reservadas para iteraciones del lenguaje interpretado, a través de la lectura y análisis de sus conceptos, observación de ejemplos y la experimentación en un entorno interactivo de ejecución, para escribir sentencias de código que efectúen iteraciones de acuerdo a valores de las variables del sistema y resulten en la ejecución de operaciones específicas, con actitud decidida y propositiva.</p>	<p>El docente ejemplifica la forma de utilizar las palabras reservadas for, while; proporciona actividades a realizar.</p> <p>El alumno:</p> <ol style="list-style-type: none"> <li>1. Realiza una lectura con los conceptos y estructuras del uso de la sintaxis para palabras clave o reservadas for, while.</li> <li>2. Observa un video instruccional para reforzar las lecturas del tema.</li> <li>3. Realiza operaciones con operadores de comparación y utiliza operadores lógicos para concatenarlos, utilizando un entorno interactivo de ejecución de código.</li> </ol> <p>Responde cuestionario en línea del tema de declaraciones con for y while.</p>	<p>Pintarrón, Equipo Audiovisual, Equipo de cómputo, documentación del lenguaje, archivos de ayuda audiovisual.</p>	2 horas
8	<p>Emplear las palabras reservadas y funciones integradas al lenguaje interpretado, por medio de la lectura de su sintaxis y la experimentación en un ambiente interactivo de ejecución, para mejorar la habilidad de redactar líneas de código reducidas al integrar buenas prácticas de codificación a las aplicaciones, de manera clara y ordenada.</p>	<p>El docente ejemplifica la forma de utilizar las funciones integradas range, enumerate, zip, in, min, max, random library, shuffle, randint, input, y proporciona actividades a realizar.</p> <p>El alumno:</p> <ol style="list-style-type: none"> <li>1. Realiza una lectura con los conceptos y estructuras del uso de la sintaxis para las funciones integradas range, enumerate, zip, in, min, max, random library, shuffle, randint, input.</li> <li>2. Observa un video</li> </ol>	<p>Pintarrón, Equipo Audiovisual, Equipo de cómputo, documentación del lenguaje, archivos de ayuda audiovisual.</p>	2 horas

		<p>instruccional para reforzar las lecturas del tema.</p> <p>3. Realiza operaciones con range, enumerate, zip, in, min, max, random library, shuffle, randint, input, utilizando un entorno interactivo de ejecución de código.</p> <p>Entrega reporte de actividades en formato electrónico.</p>		
9		<p>El docente explica el concepto de comprensión de listas (comprehension list) y proporciona actividades a realizar.</p> <p>El alumno:</p> <ol style="list-style-type: none"> <li>1. Observa un video instruccional para reforzar los conceptos explicados por el docente.</li> <li>2. Realiza ejercicios con listas utilizando un entorno interactivo de ejecución de código.</li> </ol> <p>Contesta un cuestionario en línea de los temas palabras reservadas y funciones integradas.</p>	<p>Pintarrón, Equipo Audiovisual, Equipo de cómputo, documentación del lenguaje, archivos de ayuda audiovisual.</p>	2 horas
10		<p>El docente explica el concepto de métodos (por ejemplo, append, pop, insert, reverse) y su uso en listas; proporciona actividades a realizar.</p> <p>El alumno:</p> <ol style="list-style-type: none"> <li>1. Observa un video instruccional para reforzar los conceptos explicados por el docente.</li> </ol>	<p>Pintarrón, Equipo Audiovisual, Equipo de cómputo, documentación del lenguaje, archivos de ayuda audiovisual.</p>	2 horas

		<p>2. Realiza ejercicios con métodos en listas utilizando un entorno interactivo de ejecución de código.</p> <p>3. Busca métodos en la documentación y elabora un ejemplo de su uso.</p> <p>Entrega un reporte con un ejemplo de creación propia del uso de un método en formato electrónico.</p>		
<b>UNIDAD III</b>				
11	<p>Emplear funciones en lenguaje interpretado, por medio de la discusión de su sintaxis y la experimentación en un ambiente interactivo de ejecución, para redactar funciones que puedan ser ejecutadas repetidas veces sin la reescritura de sus líneas de código, con determinación y creatividad.</p>	<p>El docente explica la sintaxis y el comportamiento de las funciones en lenguaje interpretado; proporciona puntos de discusión y modera las opiniones.</p> <p>El alumno:</p> <ol style="list-style-type: none"> <li>1. Participa en el foro de discusión en el taller.</li> <li>2. Propone ejemplos de funciones.</li> <li>3. Realiza un compendio con los ejemplos y verifica su funcionalidad en un ambiente de desarrollo interactivo.</li> </ol> <p>Entrega un reporte con los productos de la sesión en formato electrónico.</p>	<p>Pintarrón, Equipo Audiovisual, Equipo de cómputo, documentación del lenguaje, archivos de ayuda audiovisual.</p>	2 horas
12	<p>Construir aplicaciones con lenguaje interpretado, por medio del uso de sus tipos de datos, palabras reservadas, funciones y métodos, para obtener productos que sean desarrollados con simplicidad, versatilidad y rapidez, de</p>	<p>El docente proporciona las especificaciones de las aplicaciones a desarrollar, que incluyen diversos tipos de datos, palabras reservadas, funciones y métodos.</p>	<p>Pintarrón, Equipo Audiovisual, Equipo de cómputo, documentación del lenguaje, archivos de ayuda audiovisual.</p>	4 horas

	manera creativa e innovadora.	<p>El alumno:</p> <ol style="list-style-type: none"> <li>1. Analiza las especificaciones y determina la cantidad y tipo de variables a utilizar, anotando sus conclusiones.</li> <li>2. Redacta un algoritmo de la secuencia de acciones que resolverán las especificaciones solicitadas.</li> <li>3. Redacta código en lenguaje interpretado y verifica su funcionalidad en un ambiente de desarrollo interactivo.</li> </ol> <p>Entrega un reporte con los productos de la sesión en formato electrónico.</p>		
13		<p>El docente muestra utilizando ejemplos funcionales el uso de los parámetros *args y **kwargs, y solicita actividades a realizar que demuestren el tema.</p> <p>El alumno:</p> <ol style="list-style-type: none"> <li>1. Analiza las especificaciones y determina la cantidad y tipo de variables a utilizar, anotando sus conclusiones.</li> <li>2. Redacta un algoritmo de la secuencia de acciones que resolverán las especificaciones solicitadas.</li> <li>3. Redacta código en</li> </ol>	Pintarrón, Equipo Audiovisual, Equipo de cómputo, documentación del lenguaje, archivos de ayuda audiovisual.	2 horas

		<p>lenguaje interpretado y verifica su funcionalidad en un ambiente de desarrollo interactivo.</p> <p>Entrega un reporte con los productos de la sesión en formato electrónico.</p>		
14		<p>El docente proporciona las especificaciones de las aplicaciones a desarrollar, que incluyen diversos tipos de datos, palabras reservadas, funciones, métodos y argumentos; proporciona las soluciones una vez terminadas las actividades del alumno.</p> <p>El alumno:</p> <ol style="list-style-type: none"> <li>4. Analiza las especificaciones y determina la cantidad y tipo de variables a utilizar, anotando sus conclusiones.</li> <li>5. Redacta un algoritmo de la secuencia de acciones que resolverán las especificaciones solicitadas.</li> <li>6. Redacta código en lenguaje interpretado y verifica su funcionalidad en un ambiente de desarrollo interactivo.</li> </ol> <p>Entrega un reporte con los productos de la sesión en formato electrónico.</p>	<p>Pintarrón, Equipo Audiovisual, Equipo de cómputo, documentación del lenguaje, archivos de ayuda audiovisual.</p>	4 horas
15	<p>Crear aplicaciones en lenguaje interpretado, utilizando como</p>	<p>El docente proporciona los enunciados que describen las</p>	<p>Pintarrón, Equipo Audiovisual, Equipo de</p>	4 horas

	<p>herramientas el ambiente de desarrollo y la documentación del lenguaje, para convertir los enunciados que describen los problemas a resolver en código óptimo y funcional, de manera ordenada y creativa,</p>	<p>necesidades a resolver y proporciona asesoría. El alumno: 1. Analiza las especificaciones y redacta un algoritmo que convierte en código en lenguaje interpretado y verifica su funcionalidad. Entrega una aplicación ejecutable y un reporte con los productos de la sesión en formato electrónico.</p>	<p>cómputo, documentación del lenguaje, archivos de ayuda audiovisual.</p>	
--	--	---	--	--

## VII. MÉTODO DE TRABAJO

**Encuadre:** El primer día de clase el docente debe establecer la forma de trabajo, criterios de evaluación, calidad de los trabajos académicos, derechos y obligaciones docente-alumno.

### **Estrategia de enseñanza (docente)**

Explicar conceptos y ejemplos, utilizar herramientas tecnológicas, propiciar el trabajo en equipo, promover el autoaprendizaje, proporciona las referencias bibliográficas, elaborar y aplicar evaluaciones utilizando herramientas tecnológicas.

### **Estrategia de aprendizaje (alumno)**

Leer, analizar e investigar, trabajar en equipo con los compañeros de clase, desarrollar la evidencia de desempeño, desarrollar diversas estrategias de aprendizaje como cuadros comparativos, lluvias de ideas, trabajo colaborativo, y exposiciones.

## VIII. CRITERIOS DE EVALUACIÓN

La evaluación será llevada a cabo de forma permanente durante el desarrollo de la unidad de aprendizaje de la siguiente manera:

### **Criterios de acreditación**

- Para tener derecho a examen ordinario y extraordinario, el estudiante debe cumplir con los porcentajes de asistencia que establece el Estatuto Escolar vigente.
- Calificación en escala del 0 al 100, con un mínimo aprobatorio de 60.

### **Criterios de evaluación**

- Evaluaciones parciales.....20%
- Productos de actividades de taller.....30%
- Tareas.....10%
- Evidencia de desempeño.....40%

(Portafolio de evidencias con la solución a problemas resueltos mediante el desarrollo de aplicaciones computacionales)

**Total...100%**

## IX. REFERENCIAS

### Básicas

Chazallet, S. (2016). *Python 3: los fundamentos del lenguaje*. (2ª ed.) España: Ediciones ENI.

Marvin, R., Ng'ang'a, M., y Omondi, A. (2018). *Python Fundamentals*. Reino Unido: Packt Publishing Limited.

Welcome to Python.org. (2019). *Python*. Recuperado de <https://www.python.org/>

### Complementarias

Jackson, C. (2018). *Learn Programming in Python with Cody Jackson*. Birmingham. Reino Unido: Packt Publishing Limited.

Lutz, M. (2014). *Python*. Reino Unido: O'Reilly. [clásica]

## X. PERFIL DEL DOCENTE

El docente que imparta esta asignatura debe contar con título de Ingeniero en Computación o área afín, debe poseer el grado de maestría y preferentemente doctorado en ciencias o ingeniería.

Es deseable el contar con experiencia profesional comprobable en el área de Ingeniería de software, así como haber acreditado cursos de formación docente y capacitación en la enseñanza y evaluación por competencias. Es indispensable ser competente en la operación de instrumentos de laboratorio y contar con amplio dominio de las TIC.

Para el desarrollo de la actividad docente en esta asignatura es necesario contar con la capacidad para interpretar información técnica en inglés. Se requiere cuenta con la habilidad de comunicación efectiva y liderazgo para propiciar el trabajo en equipo. Adicionalmente, ser una persona proactiva, analítica, responsable, con un alto sentido de la ética y con vocación de servicio a la enseñanza.