

Guía de Trabajo

Algoritmia

Miguel Tupac Yupanqui Alanya

Guía de Trabajo

Algoritmia

Material publicado con fines de estudio.

Código: (...)

Huancayo, 2023

De esta edición

© Universidad Continental, Oficina de Gestión Curricular Av. San Carlos 1795,
Huancayo-Perú

Teléfono: (51 64) 481-430 anexo 7361

Correo electrónico: recursosucvirtual@continental.edu.pe

<http://www.continental.edu.pe/>

Cuidado de edición Fondo Editorial

Diseño y diagramación Fondo Editorial

Todos los derechos reservados.

La *Guía de Trabajo*, recurso educativo editado por la Oficina de Gestión Curricular, puede ser impresa para fines de estudio.

Contenido

Presentación	5
Primera Unidad	7
Introducción a la resolución de problemas con computadoras	
Semana 1: Sesión 2	
Instalación de Python y VS Code	8
Semana 2: Sesión 2	
Algoritmos	9
Semana 3: Sesión 2	
Introducción a la programación: Identificadores, datos, variables	10
Semana 4: Sesión 2	
Introducción a la programación: Expresiones, entradas y salidas	12
Segunda Unidad	13
Estructuras de control	
Semana 5: Sesión 2	
estructuras de control secuenciales	14
Semana 6: Sesión 2	
Estructura de control selectivas	15
Semana 7: Sesión 2	
Estructuras de control repetitivas	16
Semana 8: Sesión 2	

Estructura de control: Secuenciales-Selectivas - Repetitivas	17
Tercera Unidad	19
Semana 9: Sesión 2	
Cadenas	20
Semana 10: Sesión 2	
Listas - Tuplas	
Semana 11: Sesión 2	
Diccionarios	
Semana 12: Sesión 2	
Estructuras de datos básicas	
Cuarta Unidad	27
Funciones y modularidad	
Semana 13: Sesión 2	28
Funciones	
Semana 14: Sesión 2	
Modularidad	29
Semana 15: Sesión 2	
Recursividad	30
Semana 16: Sesión 2	
Programación modular	31
Referencias	32

Presentación

Esta guía de trabajo es fundamental para respaldar el aprendizaje de los contenidos de la asignatura de Algoritmia. En ella, encontrarás una variedad de actividades prácticas destinadas a facilitar la comprensión y aplicación de los conceptos teóricos abordados en clase. Estos incluyen la resolución de problemas con computadoras, algoritmos, lenguajes de programación, introducción a la programación, estructuras de control (secuenciales, selectivas, repetitivas), el manejo de estructuras de datos como cadenas, listas, tuplas y diccionarios, concluyendo con funciones, modularidad, uso de módulos y bibliotecas, y la técnica de la recursividad. Estos conocimientos son esenciales para desarrollar programas computacionales de manera eficiente.

El resultado de aprendizaje de esta asignatura implica la capacidad del estudiante para elaborar programas para computadoras empleando las estructuras de control apropiadas y entendiendo el uso de módulos de programa. La asignatura se divide en cuatro unidades esenciales: Unidad 1 - Introducción a la Resolución de Problemas con Computadoras, estableciendo las bases conceptuales necesarias para abordar sistemáticamente la resolución de problemas. Unidad 2 - Estructuras de Control: Secuencial – Selectiva - Repetitiva, explorando las estructuras de control fundamentales que definen la lógica de un programa. Unidad 3 - Estructuras de Datos Básicas, que se centra en la manipulación de datos y estructuras como cadenas, listas, tuplas y diccionarios. Unidad 4 - Funciones y Modularidad, donde se profundiza en la creación de funciones y su aplicación en la programación modular.

Para aprovechar al máximo esta guía, es esencial que los estudiantes se involucren activamente en las prácticas y ejercicios propuestos. Se sugiere mantener una rutina de estudio constante, resolver problemas de programación de forma regular y buscar la retroalimentación del docente. Además, explorar recursos adicionales como tutoriales en línea y ejemplos de código puede enriquecer su comprensión de los conceptos. La paciencia y la persistencia son claves en el proceso de aprender a programar, por lo que es importante plantear dudas y enfrentar desafíos sin temor.

Miguel Tupac Yupanqui A.

Primera **Unidad**

**Introducción a la resolución de
problemas con computadoras**

Semana 1: Sesión 2

Instalación de Python y VS Code

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 1

Nombres y apellidos:

Instrucciones

- Lee la guía con atención, esto te dará una visión general de lo que se espera.
- Realiza los pasos en el orden indicado, es importante para que la actividad se realice de manera efectiva y sin problemas.

I. Propósito

El objetivo de esta actividad es enseñarte cómo instalar Python en tu computadora. Esta guía ofrece instrucciones detalladas para descargar e instalar Python y VS Code en un sistema operativo Windows. Una vez completada las instalaciones, podrás crear y ejecutar tus propios programas.

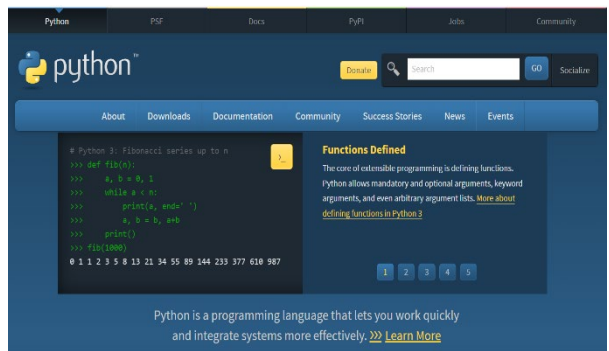
II. Descripción de la actividad por realizar

Instalación de Python

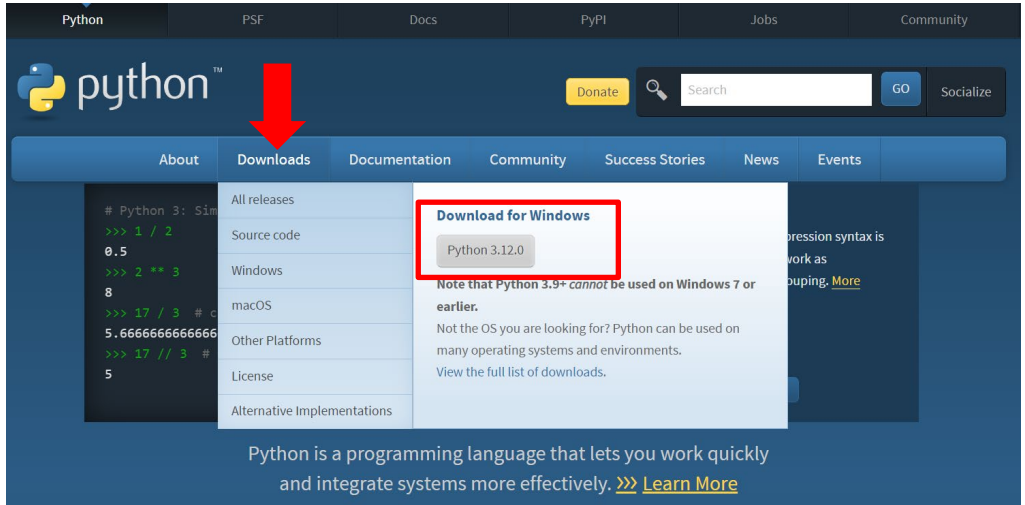
1. Abra su navegador web y vaya al sitio web oficial de Python:

<https://www.python.org/>

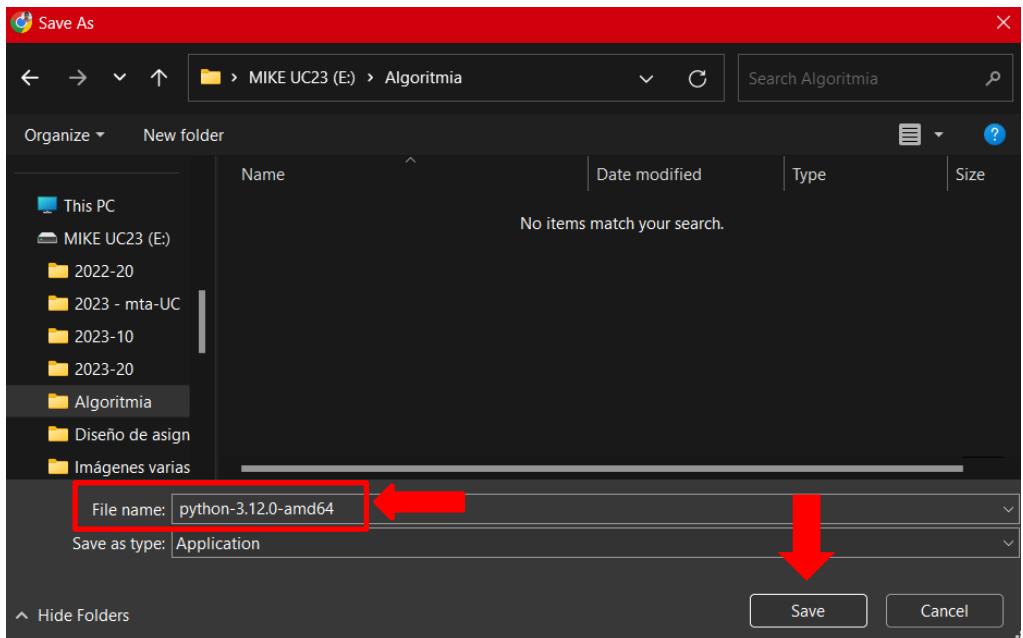
Nota: tomada de <https://www.python.org/>



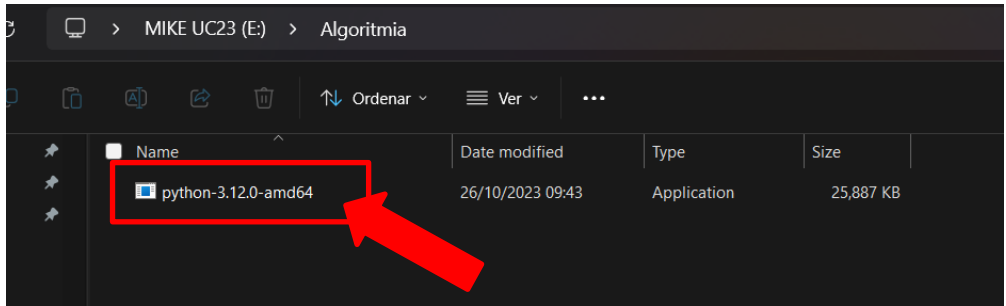
2. En la página de descargas de Python, verá la última versión estable disponible. Haga clic en Download for Windows "**Python 3.12.0**" (es la versión actual al 26/10/23) para descargar el instalador.



3. Se descargará un archivo ejecutable (por ejemplo, **python-3.12.0-amd64.exe**).

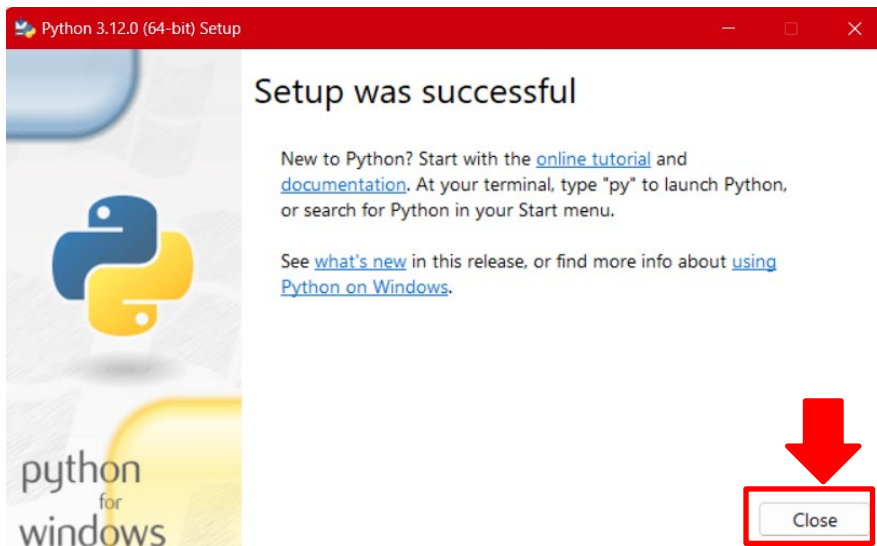
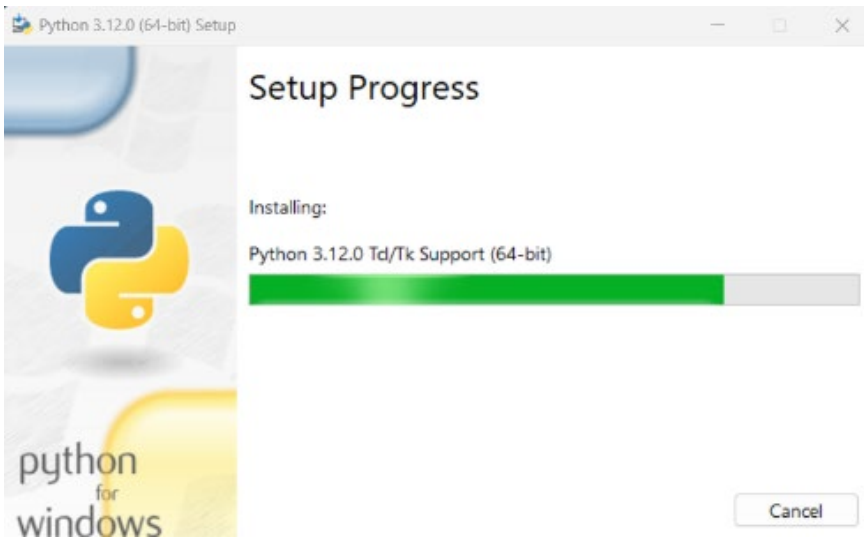


4. Haz doble clic en el archivo descargado para abrir el asistente de instalación.



5. Sigue las indicaciones en la pantalla para finalizar la instalación. Asegúrate de seleccionar la opción "Agregar Python 3.12 al PATH" para acceder a Python desde cualquier ubicación.





6. Para verificar que Python se haya instalado correctamente, abre una ventana de símbolo del sistema y escribe el comando `python --version`.

```
C:\> Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.3448]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\user>python --version
```

7. Deberías ver algo como esto:

```

C:\> Símbolo del sistema

Microsoft Windows [Versión 10.0.19045.3448]
(c) Microsoft Corporation. Todos los derechos reservados.

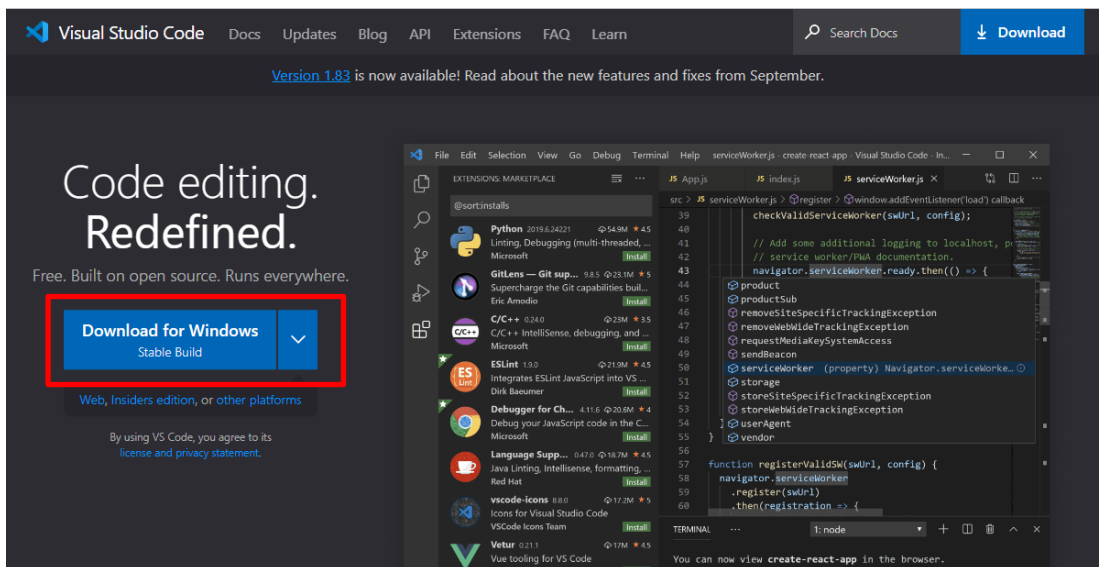
C:\Users\user>python --version
Python 3.12.0

```

8. ¡Listo! ya tienes Python instalado en tu computadora y estás preparado para empezar a programar.

Instalación de Visual Studio Code

1. Descargar el instalador de Visual Studio Code desde el sitio web oficial: <https://code.visualstudio.com/>
2. Seleccione y descargue el instalador de Visual Studio Code para Windows.



Después de descargar el instalador, es posible que tenga que abrir Explorador de archivos e ir a la ubicación en la que lo instaló el explorador web. La ubicación más común es la carpeta *Descargas*.

3. Haz doble clic en el archivo del instalador para comenzar la instalación y ejecutar el programa.

Acuerdo de Licencia

Es importante que lea la siguiente información antes de continuar.

Por favor, lea el siguiente acuerdo de licencia. Debe aceptar las cláusulas de este acuerdo antes de continuar con la instalación.

Esta licencia se aplica al producto Visual Studio Code. El código fuente para Visual Studio Code está disponible en <https://github.com/Microsoft/vscode> según el acuerdo de licencia del MIT en <https://github.com/microsoft/vscode/blob/main/LICENSE.txt>. Encontrará información adicional sobre licencias en nuestras preguntas frecuentes en <https://code.visualstudio.com/docs/supporting/faq>.

TÉRMINOS DE LICENCIA DEL SOFTWARE DE MICROSOFT

VISUAL STUDIO CODE DE MICROSOFT

- Acepto el acuerdo
- No acepto el acuerdo

Siguiente >

Cancelar

Seleccione las Tareas Adicionales

¿Qué tareas adicionales deben realizarse?

Seleccione las tareas adicionales que desea que se realicen durante la instalación de Visual Studio Code y haga clic en Siguiente.

Accesos directos adicionales:

- Crear un acceso directo en el escritorio

Otros:

- Agregar la acción "Abrir con Code" al menú contextual de archivo del Explorador de Windows
- Agregar la acción "Abrir con Code" al menú contextual de directorio del Explorador de Windows
- Registrar Code como editor para tipos de archivo admitidos
- Agregar a PATH (disponible después de reiniciar)

Listo para Instalar

Ahora el programa está listo para iniciar la instalación de Visual Studio Code en su sistema.

Haga clic en Instalar para continuar con el proceso o haga clic en Atrás si desea revisar o cambiar alguna configuración.

Tareas Adicionales:
 Accesos directos adicionales:
 Crear un acceso directo en el escritorio
 Otros:
 Agregar la acción "Abrir con Code" al menú contextual de archivo del Explorador de Windows
 Agregar la acción "Abrir con Code" al menú contextual de directorio del Explorador de Windows
 Agregar a PATH (disponible después de reiniciar)

< Atrás

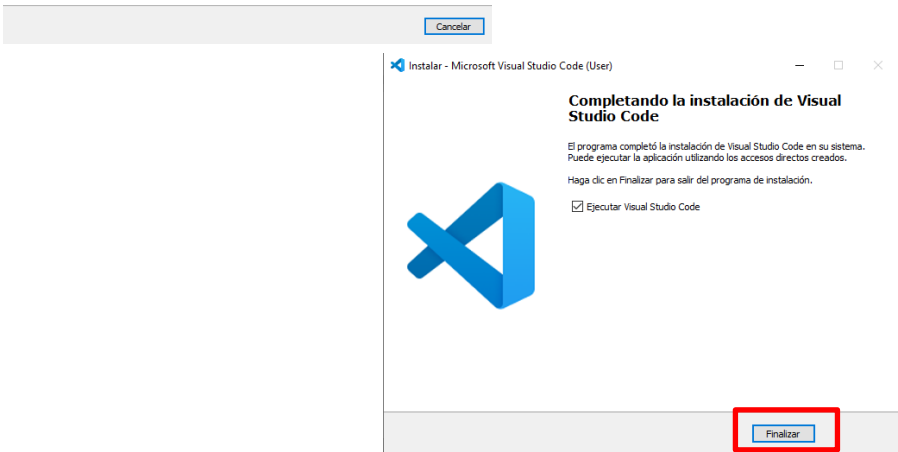
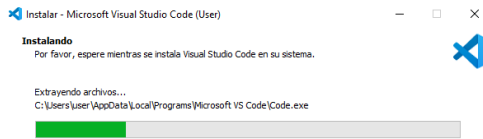
Siguiente >

Cancelar

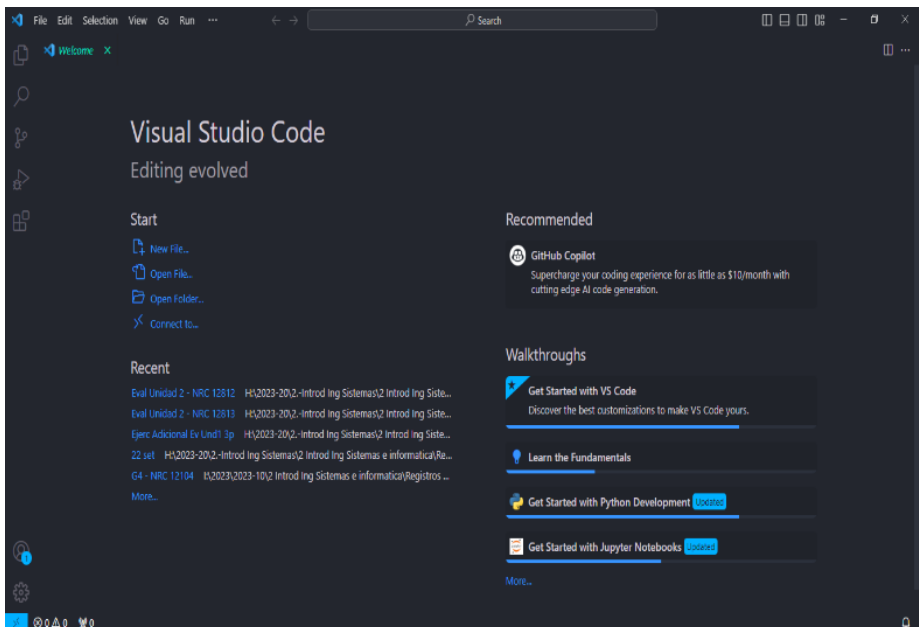
< Atrás

Instalar

Cancelar



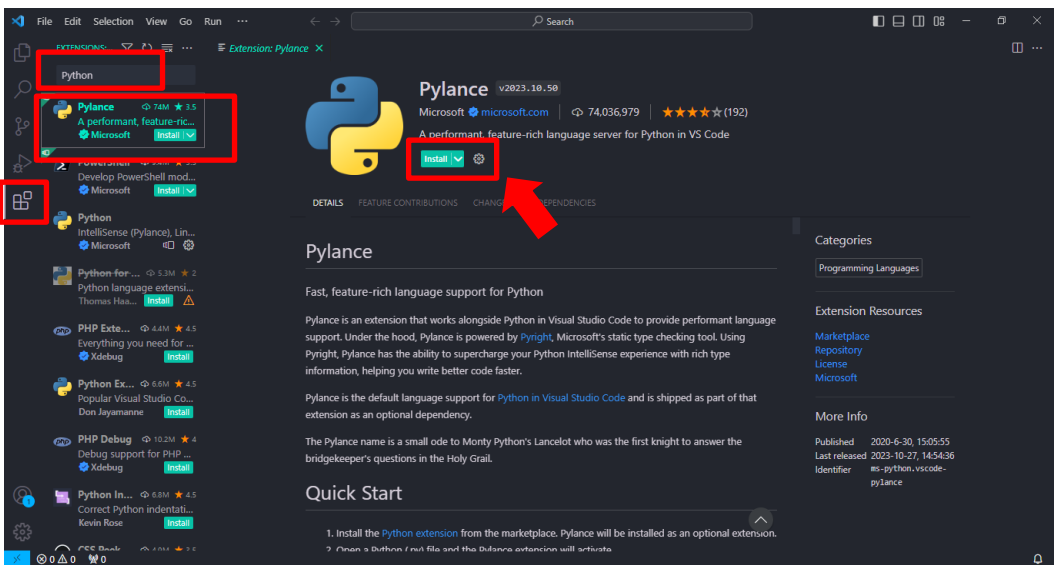
- Una vez completada la instalación, Visual Studio Code se inicia automáticamente.



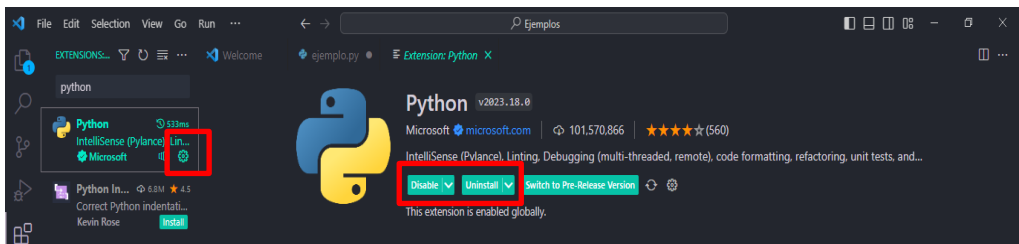
Instalación de la extensión de Python para Visual Studio Code

Visual Studio Code es un editor de código que ofrece soporte para varios lenguajes de programación. La extensión de Python para Visual Studio Code brinda características visuales como resaltado de sintaxis y autocompletado, además de herramientas de depuración que mejoran y aceleran la escritura de código en Python.

1. Para filtrar la lista de extensiones, escriba python en el cuadro de búsqueda situado en la parte superior de la vista Extensiones.
2. Seleccione la extensión Python publicada por Microsoft, descrita como IntelliSense (Pylance), y normalmente la primera de la lista. Los detalles sobre esa extensión aparecen en un panel con pestañas a la derecha.
3. En el panel Extensiones o en el panel principal, seleccione Instalar.

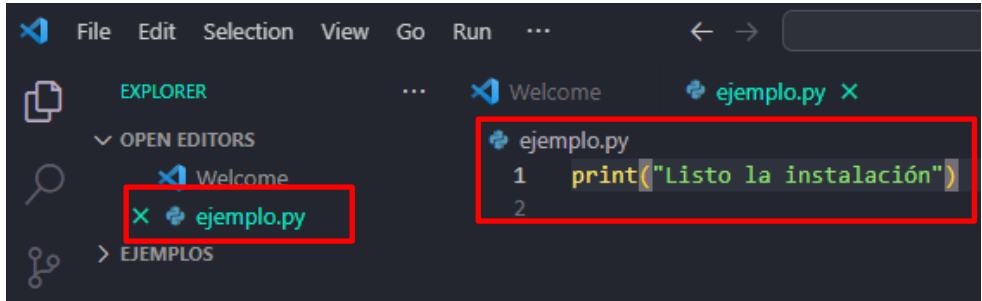


4. Cuando finaliza la instalación, el botón Instalar cambia a un icono de Configuración y en la vista Extensiones a cambiado a dos botones, Deshabilitar y Desinstalar, en el panel principal.

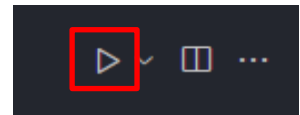


Probar si ya lo tienes todo listo para programar en Python

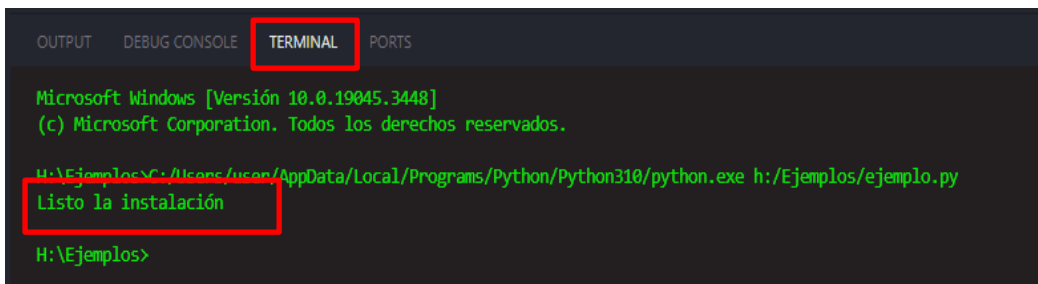
1. Para comprobar si tienes todo el procedimiento realizado hasta ahora correctamente, y que puedes empezar a programar en Python, escribe la siguiente línea de código en el archivo que acabas de crear:



2. Para ejecutar el código que has escrito, haz click en este botón:



3. Abajo, se abrirá la consola y te dirá si tienes algún error o ha ido todo bien. Si te aparece el mensaje, es que todo ha ido correcto.



Semana 2: Sesión 2

Algoritmos

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 1

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándote de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El objetivo de esta actividad es crear algoritmos que resuelvan problemas de manera eficiente. Para lograrlo, es fundamental emplear un enfoque lógico y creativo en la búsqueda de soluciones.

II. Descripción de la actividad por realizar

Para los problemas que se plantean a continuación, debe representar los algoritmos mediante diagramas de flujo o pseudocódigo.

1. Implementar un algoritmo que determine el área y el perímetro de un triángulo rectángulo a partir de la medida de su base y altura.
2. Implementar un algoritmo que calcule el promedio ponderado de una asignatura, los pesos de cada rubro son: 20% Prácticas, 25% Laboratorios, 20% Teoría y 35% Examen final.
3. Implementar un algoritmo que transforme una cantidad de grados Celsius a grados Fahrenheit. ($Fahrenheit = (Celsius * 9/5) + 32$)

4. Implementar un algoritmo para determinar el IGV (18%) del monto total de compra de un determinado producto. Deberá mostrar precio de venta, monto del IGV, precio total.
5. Implementar un algoritmo que determine si un número entero es impar.
6. Un establecimiento comercial en el centro de la ciudad brinda descuento del 20% a los clientes que compran más de S/ 250.00 Soles. Implementar un algoritmo que determine si un cliente tiene derecho al descuento y, en caso afirmativo, calcule el importe del mismo.
7. Implementar un algoritmo que sume todos los números pares en un rango dado. El usuario debe proporcionar el valor inicial y final del rango.
8. Implementar un algoritmo que solicite al usuario ingresar tres números enteros y devuelva como salida los números ordenados de menor a mayor.
9. Implementar un algoritmo que solicite al usuario ingresar una secuencia de números enteros entre positivos y negativos, hasta que se ingrese un cero. Luego, deberá mostrar la suma y el promedio de los números positivos y negativos ingresados.
10. Implementar un algoritmo que lea la edad de 8 personas, para luego mostrar la edad menor y mayor de todas edades ingresadas.

Semana 3: Sesión 2

Introducción a la programación

Identificadores, datos, variables, constantes, comentarios, operadores

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 1

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándose de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El propósito de esta actividad es fortalecer la comprensión de conceptos clave de programación, lo que capacitará a los estudiantes para afrontar desafíos y proyectos de mayor complejidad.

II. Descripción de la actividad por realizar

A. Con respecto a los identificadores responder:

1. ¿Cuál de las siguientes opciones es un identificador válido en Python?
 - a) 123identifier
 - b) mi_identificador
 - c) identificador#1
 - d) \$identificador

2. ¿Cuál de las siguientes opciones es un identificador inválido en Python?
- a) total_score
 - b) Mi_Variable
 - c) while
 - d) MyValue2
3. ¿Cuál es la regla para nombrar identificadores en Python?
- a) Pueden empezar con cualquier carácter, incluyendo números.
 - b) No pueden contener guiones bajos (_)
 - c) Solo se permiten letras mayúsculas en los identificadores.
 - d) Deben comenzar con una letra o guion bajo (_), seguido de letras, números o guiones bajos.
4. ¿Qué opción considera una buena práctica para nombrar identificadores en Python?
- a) Utilizar nombres largos y descriptivos para mejorar la legibilidad.
 - b) Evitar el uso de palabras clave reservadas de Python como identificadores.
 - c) Empezar los identificadores con un número para que sean más fáciles de recordar.
 - d) Utilizar solo letras mayúsculas para destacar los identificadores importantes.
5. ¿Cuál de las siguientes opciones es un identificador válido en Python?
- a) 2_my_variable
 - b) variable@
 - c) variable_2
 - d) break

Respuestas: 1. (b) 2.(c) 3.(d) 4.(b) 5.(c)

B. Resolver los siguientes ejercicios:

1. ¿Cuáles serán los resultados al evaluar cada una de las siguientes expresiones en Python? (se recomienda realizar manualmente el cálculo de cada expresión, luego con la asistencia del ordenador, verifica si tu resultado es correcto).

a) $4 + 3 * 2 + 5$

c) $-9 / 2 - 2$

b) $(6 + 3) * 4 + 8$

d) $-9 // 2 - 1$

2. Minimizando el uso de paréntesis, transforma las siguientes expresiones matemáticas a Python y realiza su evaluación.

a) $6 + (5^{(6/3)})$

c) $-(4^3)^2$

b) $(6/3)^4 + (12/2)^2$

d) $\frac{12 + 2^3}{3^3 - 2}$

3. Determinar el resultado tras ejecutar las líneas de código siguientes:

a)

```
>>> x = 10
>>> x = x * 10
>>> x
```

b)

```
>>> a = 5
>>> b = 10
>>> a = a * b
>>> b = a % 3
>>> c = a + b
>>> c
```

c)

```
>>> z = 2
>>> z += 2
>>> z += 2 - 2
>>> z *= 2
>>> z *= 1 + 1
>>> z //= 2
>>> z %= 3
>>> z /= 3 - 1
>>> z -= 2 + 1
>>> z -= 2
>>> z **= 3
>>> z
```

4. Si: $z = 'y'$ Determina el resultado tras evaluar las expresiones y sentencias siguientes. El resultado obtenido, compara con la respuesta proporcionada por el intérprete de Python.

a)

```
>>> z + 'y'
```

b)

```
>>> z + 'z'
```

c) <pre>>>> z * 3 + 'y' * 4</pre>	d) <pre>>>> 3 * (z + 'y')</pre>
e) <pre>>>> 4 * ('z' + 'y')</pre>	f) <pre>>>> 6 * (z + 'y' + '4')</pre>

5. Determina el resultado tras evaluar las expresiones y sentencias siguientes. El resultado obtenido, compara con la respuesta proporcionada por el intérprete de Python.

```
a)
>>> 'z' * 4 + '/' * 3 + 2 * 'xyz' + '+='
```

```
b)
>>> nombre_palindromo = 'Otto'
>>> (6 * '=' + nombre_palindromo + '=' * 6) * 3
```

```
c)
>>> subcadena = '#' + '$' * 4 + '#'
>>> '6' * 5 + 3 * subcadena + 5 * '9'
```

```
d)
>>> 2 * '5' + '.' + '3' * 2 + 'e-' + 4 * '69'
```

6. Analiza los patrones en las siguientes cadenas y crea expresiones que, desde subcadenas más cortas y empleando operadores de concatenación y repetición, generen las cadenas mostradas. Si crees conveniente, utiliza variables para la creación de las expresiones.

```
a)
'$$$$$/#/#/#<<=>><<=>>'
```

```
b)
')@()@()@()@(* ***)@()@()@()@(* ***)'
```

```
c)
'ottootottoototto*-*-*-*-////ottoototto????'
```

```
d)
'/*/*/*/*/*/*+++++---+++++/*/*/*/*/*/*--'
```

Semana 4: Sesión 2

Introducción a la programación

Expresiones, entradas y salidas

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 1

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándote de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El objetivo de esta actividad es consolidar la comprensión de los conceptos básicos de programación, permitiendo a los estudiantes adquirir las destrezas necesarias para abordar desafíos y proyectos de mayor complejidad.

II. Descripción de la actividad por realizar

A. Resolver los siguientes ejercicios:

1. Implemente un programa que solicite al usuario su nombre y edad, almacenándolos en las variables "nombre" y "edad". Posteriormente, muestra un mensaje que diga: "Hola {nombre}, tienes {edad} años"
2. Implemente un programa que, al introducir el radio de un círculo, calcule su área. Utiliza la constante PI con el valor 3.1416 y guarda el resultado en la variable "area". Posteriormente, muestra el valor de "area" con dos decimales

- Implementa un programa que, al recibir un número entero del usuario, determine si es par o impar. Utiliza el operador % para obtener el resto de una división. El programa debe mostrar como salida el mensaje: "El número {número} es par" o "El número {número} es impar"
- Implemente un programa que compare dos números enteros ingresados por el usuario y determine cuál es mayor, menor o si son iguales. Utiliza los operadores relacionales >, < y == para realizar la comparación. Posteriormente, muestra un mensaje que indique: "{número1} es mayor que {número2}", "{número1} es menor que {número2}" o "{número1} es igual a {número2}".

- Implemente un programa que, dados dos números enteros "a" y "b", ambos mayores a cero, calcule el valor de "Z".

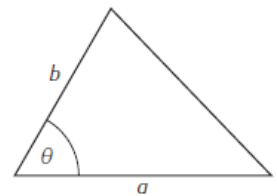
$$Z = \sqrt{\frac{3a}{7+b}} + \left(\frac{5b^3}{2+a}\right)^2$$

- Implemente un programa que, dados dos números enteros "a" y "b", ambos superiores a cero, calcule el valor de "Z".

$$X = \left(\frac{64a}{2b^2+a}\right)^3 + \sqrt{\frac{5b}{2+a}}$$

- Implemente un programa que solicite los valores de los tres lados de un triángulo y, a partir de ellos, calcule tanto el área como el perímetro. Tener en consideración que el área de un triángulo a partir de sus tres lados (a, b y c) puede calcularse de la siguiente forma: $A = \sqrt{s(s-a)(s-b)(s-c)}$ donde: $s = (a + b + c)/2$.

- Implemente un programa que solicite al usuario el valor de dos de sus lados (a, b), y el valor del ángulo θ que estos forman entre sí, a partir de ellos calcule su área. Tener en consideración que para calcular el área puede usar la fórmula: $A = \frac{1}{2} ab \sin(\theta)$



Considera que la función seno (sin) en Python opera en radianes. Por lo tanto, al ingresar un ángulo en grados, es necesario convertirlo a radianes, sabiendo que π radianes equivalen a 180 grados. (Ejm.: $a = 1$, $b = 2$, $\theta = 30$; el resultado es 0.5).

Segunda **Unidad**

Estructuras de control

Semana 5: Sesión 2

Estructuras de control Secuenciales

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 2

Nombres y apellidos:

Instrucciones

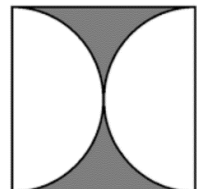
Lea detenidamente cada enunciado asegurándote de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El propósito de esta actividad es implementar programas computacionales que empleen estructuras de control secuencial para su solución.

II. Descripción de la actividad por realizar

1. Implementa un programa que permita introducir una medida en pulgadas y muestre su equivalente en centímetros, considerando que 1 pulgada equivale a 2,54 centímetros.
2. Implementar un programa que, a partir del lado del cuadrado ingresado por el usuario, determine el área sombreada (ver figura). Considerar el valor de $\pi = 3.1416$
3. Implementar un programa que calcule el porcentaje de hombres y mujeres dentro de un grupo de estudiantes



4. Implementar un programa que permita a un micro empresario distribuir una cantidad de dinero entre las cinco áreas que posee, de acuerdo a la siguiente distribución:
 - Gerencia General: 40% del monto recibido por RRHH.
 - RRHH: 35% de la cantidad a repartir.
 - Marketing: 23% del monto total recibido entre RRHH y Operaciones.
 - Operaciones: 25% de la cantidad a repartir.
 - Servicios generales: Lo que queda del dinero repartir

5. En el mundo mágico de Harry Potter, los alumnos de Hogwarts están aprendiendo sobre encantamientos espaciales. Hermione, fascinada con la magia matemática, ha descubierto un antiguo pergamino que contiene fórmulas para calcular distancias en el plano cartesiano con precisión mágica. Ayuda a Hermione a desarrollar un programa encantado que permita a los alumnos calcular la distancia entre dos puntos del plano cartesiano para mejorar sus habilidades mágicas en la asignatura de Encantamientos Espaciales.

6. En el Hospital Princeton-Plainsboro, el Dr. House y su equipo están investigando la relación entre el ejercicio y las pulsaciones cardíacas. Durante una sesión de entrenamiento, el Dr. House desafía a su equipo a crear un programa que calcule el número de pulsaciones que una persona debe tener por cada 10 segundos de ejercicio, utilizando una fórmula específica: ***Num_Pulsaciones = (220 – edad)/10*** Ayuda al equipo del Dr. House a desarrollar este programa para mejorar la comprensión de la relación entre el esfuerzo físico y las pulsaciones cardíacas.

7. En la sede gubernamental de Finanzas, el Ministro de Economía ha planteado un desafío al equipo de analistas financieros: desarrollar un programa que calcule el monto final que un cliente debe pagar después de recibir un descuento del 15% sobre el total de su compra en una tienda específica. Este programa ayudará a los ciudadanos a comprender mejor el impacto del descuento en sus compras. Ayuda al equipo gubernamental a crear este programa para fomentar la educación financiera entre los ciudadanos.

8. En el Hospital Princeton-Plainsboro, el Dr. House ha propuesto un desafío a su equipo médico: desarrollar un programa informático que distribuya de manera precisa el presupuesto anual entre las tres áreas principales del hospital: Ginecología, Pediatría y Traumatología. Según

la distribución establecida, el 40% del presupuesto se destina a Ginecología, el 30% a Traumatología y otro 30% a Pediatría. Este programa ayudará a los administradores y médicos a calcular con exactitud la cantidad de financiamiento que recibirá cada área, independientemente del monto total asignado. Ayuda al equipo del Dr. House a crear esta herramienta para mejorar la gestión de los recursos del hospital.

9. En la ciudad de Piedradura, Pedro Picapiedra, un vendedor entusiasta, busca calcular sus ingresos mensuales. Pedro recibe un sueldo base junto con un extra del 10% por comisión sobre sus ventas. Él necesita saber cuánto dinero obtendrá en concepto de comisiones por las tres ventas que realiza al mes y cuál será el total que recibirá, considerando su sueldo base y las comisiones. Ayúdalo a desarrollar un programa que le permita calcular estos ingresos mensuales y a planificar sus finanzas de manera más efectiva.
10. Estás trabajando en una aplicación de salud y necesitas calcular el Índice de Masa Corporal (IMC) de los usuarios. El IMC se calcula dividiendo el peso de la persona (en kilogramos) entre el cuadrado de su altura (en metros). El programa deberá solicitar al usuario su peso y altura, calcular el IMC y mostrar el resultado.
11. En el ficticio Hospital Mercy, el equipo médico liderado por la Dra. Grey está desarrollando una aplicación de salud avanzada. Para esta aplicación, necesitan calcular el Índice de Masa Corporal (IMC) de los pacientes. El IMC se determina con la siguiente fórmula: **$IMC = \text{Peso}(kg) / \text{Talla}^2(m)$** El programa debe pedir al usuario su peso y talla, calcular el IMC y mostrar el resultado. Esto ayudará a los pacientes a monitorear su salud desde casa. Ayuda al equipo de la Dra. Grey a implementar este cálculo en la aplicación para promover hábitos saludables entre los usuarios.

Semana 6: Sesión 2

Estructuras de control Selectivas

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 2

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándote de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El propósito de esta actividad es implementar programas computacionales que empleen estructuras de control selectivas para su solución.

II. Descripción de la actividad por realizar

1. En el hospital del ficticio Grey Sloan Memorial, el equipo médico está desarrollando un programa informático para comparar las edades de dos pacientes. El programa solicita la edad de dos personas y determina quién es más joven. En caso de que ambas tengan la misma edad, se mostrará un mensaje indicando esta situación. Esta herramienta ayudará al personal médico a manejar la información de manera más eficiente. Ayuda al equipo de Grey Sloan Memorial a crear este programa para mejorar su proceso de análisis de datos de pacientes.
2. Estás a cargo del sistema de tarifas de un servicio de transporte público. Las tarifas se calculan según la edad del pasajero de la siguiente manera:
 - Los niños menores de 12 años pagan S/ 1.0
 - Los adultos de 12 a 65 años pagan S/ 1.5

- Los adultos mayores de 65 años viajan gratis.

Debes implementar un programa que luego de ingresar la edad del pasajero y muestre la tarifa en soles que le corresponde.

3. En el mundo mágico de Harry Potter, Hermione Granger está investigando encantamientos matemáticos. Ella necesita desarrollar un programa que, al recibir un número entero, determine si este número es el doble de un número impar. Este programa será útil para comprender y aplicar conceptos mágicos matemáticos. Ayuda a Hermione a crear esta herramienta para fortalecer sus estudios sobre magia y matemáticas.
4. En la escuela de magia Hogwarts, los alumnos están aprendiendo sobre la utilidad de la magia en la vida diaria. Se necesita desarrollar un programa encantado que calcule el desglose mínimo en billetes y monedas de una cantidad exacta de Galeones. En el mundo mágico existen billetes de 200, 100, 50, 20, 10 Galeones y monedas de 5, 2 y 1 Galeón. Esta herramienta será útil para los estudiantes en su práctica diaria de encantamientos financieros. Ayuda a crear este programa encantado para facilitar el aprendizaje de los alumnos sobre manejo de monedas y billetes mágicos.
5. En el Departamento de Finanzas del Gobierno Central, se requiere un programa que calcule el monto a pagar por horas extras trabajadas en empresas. Si las horas trabajadas exceden las 40 horas semanales, el excedente se considera como horas extras. Estas se pagan al doble del salario regular si no exceden las 8 horas extras, y las primeras 8 horas extras se pagan al doble, mientras que el excedente se paga al triple del salario regular. Este programa ayudará a calcular los pagos exactos por horas extras para los trabajadores, optimizando la gestión de recursos en las empresas. Ayuda al equipo del Gobierno Central a crear esta herramienta para mejorar la equidad en el pago de horas extras a nivel nacional.
6. En el mundo literario de los libros clásicos, Sherlock Holmes, el famoso detective, se embarca en un desafío intelectual. Quiere desarrollar un programa que lea un carácter ingresado desde el teclado y le muestre a su fiel compañero, el Dr. Watson, un mensaje. Si el carácter es una letra mayúscula, mostrará "Es una MAYÚSCULA". Si es una letra minúscula, indicará "Es una MINÚSCULA". En cualquier otro caso, comunicará "NO es una letra". Ayuda a Sherlock Holmes a resolver este misterio informático para mejorar su conocimiento sobre las letras y sus características.

7. Una plataforma de streaming te ha contratado para implementar una nueva funcionalidad en su sistema. Dicha funcionalidad consiste en clasificar a los usuarios en tres categorías: "novato", "intermedio" o "avanzado", basándose en la cantidad de horas de contenido que han consumido. - Los usuarios que han consumido menos de 50 horas de contenido son "novatos". - Los usuarios que han consumido entre 50 y 200 horas son "intermedios". - Los usuarios que han consumido más de 200 horas son "avanzados". Debes crear un programa que, dadas las horas de contenido consumido por un usuario, determine su categoría correspondiente.

8. Estás trabajando en el desarrollo de un sistema de reservas para un hotel que ofrece tres tipos de habitaciones: sencillas, dobles y suites, con precios por noche de S/ 100, S/ 180 y S/ 300 respectivamente. El programa que estás creando solicitará al usuario elegir el tipo de habitación que desea reservar y la cantidad de noches. A continuación, calculará el costo total de la reserva. Si el usuario opta por reservar una suite por más de tres noches, se aplicará un descuento del 10% sobre el total de la reserva. El objetivo es que este programa permita al usuario seleccionar el tipo de habitación y la cantidad de noches para, finalmente, calcular y mostrar el costo total de la reserva, aplicando el descuento si corresponde.

9. Estás trabajando en una aplicación de fitness y necesitas calcular el gasto calórico de los usuarios según el tipo de ejercicio que realizan. Tienes la siguiente información:
 - 30 minutos de running queman 300 calorías.
 - 30 minutos de ciclismo queman 200 calorías.
 - 30 minutos de natación queman 250 calorías.El programa deberá solicitar al usuario el tipo de ejercicio y la cantidad de tiempo que lo practica, calcular el gasto calórico y mostrar el resultado.

Semana 7: Sesión 2

Estructuras de control

Repetitivas

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 2

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándote de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El propósito de esta actividad es implementar programas computacionales que empleen estructuras de control repetitivas para su solución.

II. Descripción de la actividad por realizar

1. Implemente un programa que determine la suma de todos los números desde 1 hasta n . Utiliza una estructura WHILE y luego una estructura FOR para resolver esta tarea.
2. Empleando una estructura WHILE, implementa un programa que realice una multiplicación, mediante sumas sucesivas.
3. Implementar un programa que solicite al usuario ingresar un número entero. El programa debe mostrar como salida los números múltiplos de 3 en un rango de 0 hasta el número introducido, junto con la cantidad total de estos múltiplos.
4. En el Reino de Geometrópolis, el sabio Pitágoras necesita tu ayuda. Él está investigando la secuencia de números pares y quiere crear un programa que, en honor a su descubrimiento del teorema de Pitágoras,

muestre en líneas separadas y en orden descendente todos los números pares desde 0 hasta 200, incluyendo tanto el 0 como el 200. ¿Puedes ayudar a Pitágoras a desarrollar este programa mágico para explorar los secretos de los números pares?

5. En el mundo de las matemáticas, Gauss y Euler necesitan tu ayuda. Están investigando el máximo común divisor (mcd) de dos números enteros positivos. Quieren un programa que calcule el mcd, que es el número más grande que divide exactamente a ambos números. ¿Podrías desarrollar un programa para ayudar a estos genios matemáticos a explorar los secretos del máximo común divisor?
6. En el mundo matemático de Arquímedes y Fibonacci, necesitan tu ayuda para explorar las tablas de multiplicar. Quieren un programa que calcule y muestre la tabla de multiplicar de cualquier número. Este programa mostrará el multiplicando, el multiplicador y el producto para cada operación. ¿Podrías desarrollar este programa para ayudar a estos grandes matemáticos a explorar las multiplicaciones de una manera más sistemática?
7. En el reino de las matemáticas, Euclides y Pitágoras están investigando una fórmula matemática especial. Necesitan un programa que les ayude a calcular esta fórmula mágica:

$$\sum_{i=n}^m i$$

Sin embargo, hay una regla importante: los números n y m , que serán introducidos por el usuario, deben cumplir con una condición específica. El programa no realizará ningún cálculo si n es mayor que m . En este caso, mostrará un mensaje en pantalla indicando que n debe ser menor o igual que m . ¿Podrías desarrollar este programa para ayudar a estos distinguidos matemáticos en su investigación?

8. En la ciudad de Matemágica, el Gran Mago Fibonacci ha desafiado a Gauss, el Mago de los Números, a calcular factoriales. Para ello, necesitan un hechizo poderoso que calcule el factorial de un número entero positivo. ¿Puedes crear un hechizo matemático que les permita calcular el factorial de cualquier número ingresado por los magos? El objetivo es que, con tu hechizo, estos grandes magos puedan explorar el poder de los factoriales y desvelar sus misterios matemáticos.
9. En un torneo de ajedrez, se están organizando equipos para una competición. La cantidad de combinaciones posibles de formar equipos de " m " jugadores de un conjunto de " n " jugadores es calculada mediante la fórmula combinatoria:

$$C_m^n = \binom{n}{m} = \frac{n!}{(n-m)!m!}$$

Implementa un programa que solicite los valores de n y m , y calcule la cantidad de combinaciones. Asegúrate de validar que " n " sea mayor o igual que " m ".

10. En la feria de libros de la ciudad, un bibliófilo entusiasta recorre los estantes seleccionando libros. Para asegurarse de que el cajero registre correctamente sus compras, el bibliófilo anota el precio de cada libro junto con la cantidad que ha tomado, calculando así el costo total de cada título. A medida que agrega más libros a su selección, suma estos costos hasta que decide que ha recopilado todos los libros que desea. Crea un programa que ayude a este apasionado de la lectura a calcular el costo total de sus compras en la feria de libros

Semana 8: Sesión 2

Estructuras de control **Secuenciales – Selectivas - Repetitivas**

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 2

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándote de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El propósito de esta actividad es implementar programas computacionales que empleen estructuras de control: secuenciales, selectivas y repetitivas para su solución.

II. Descripción de la actividad por realizar

1. Un cine local desea automatizar su sistema de reservas. Tienen tres tipos de clientes: estudiantes, adultos y jubilados, cada uno con un precio de entrada diferente. Además, ofrecen un descuento del 10% a los clientes que reserven más de tres entradas a la vez. Desarrolla un programa que, dada la cantidad y tipo de entradas reservadas, calcule el costo total de la reserva. El programa debe tomar en cuenta el tipo de cliente y si la cantidad de entradas reservadas es mayor a tres para aplicar el descuento.

2. La agencia de viajes "Aventuras" desea contar con un programa para registrar sus ventas el cual debe tener un menú principal con las siguientes opciones:

----- Menú Principal -----

1. Registrar venta
2. Reporte general
3. Salir

Al seleccionar la **opción 1: Registro de ventas**, debe mostrar un submenú con los destinos turísticos que ofrece:

===== Destinos Turísticos =====

- Egipto: Gran Pirámide de Guiza
- París: Torre Eiffel
- Italia: Torre de Pisa
- Volver

Al elegir un destino debe solicitar la cantidad de pasajes a comprar, género del cliente. Cada opción tiene asignado un descuento y este se aplica si la cantidad de pasajes a comprar es mayor a 3, según la siguiente tabla:

Destino turístico	Precio x persona S/	Descuento
Egipto: Gran Pirámide de Guiza	2500	10%
París: Torre Eiffel	3000	15%
Italia: Torre de Pisa	2000	20%

Deberá mostrar por venta:

<ul style="list-style-type: none"> • Género del cliente (Femenino/Masculino) • Destino seleccionado • Cantidad de pasajes a comprar 	<ul style="list-style-type: none"> • Precio del pasaje por persona S/ • Costo total de la compra S/ • Descuento en % y en S/ • Costo final con descuento S/
--	---

Al seleccionar la **opción 2: Reporte general**, debe mostrar las cantidades de:

- | | |
|--|---|
| <ul style="list-style-type: none"> • Clientes de género femenino atendidos • Pasajes vendidos • Pasajes vendidos a Egipto • Pasajes vendidos a París | <ul style="list-style-type: none"> • Pasajes vendidos a Italia • Total, acumulado de las ventas (sumatoria de los costos finales con descuento) |
|--|---|

Al seleccionar la **opción 3: Salir**, debe mostrar un mensaje:

 Gracias...
 Hasta pronto...

Tercera **Unidad**

Estructuras de datos básicas

Semana 9: Sesión 2

Cadenas

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 3

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándote de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El propósito de esta actividad es potenciar las destrezas en la manipulación de texto, incluyendo la indexación, concatenación, búsqueda, sustitución y división de cadenas. Estas habilidades son fundamentales y aplicables en diversas situaciones de programación.

II. Descripción de la actividad por realizar

1. Implementa un programa que solicite al usuario ingresar una cadena de texto. Luego, muestra como salida:
 - Los tres primeros caracteres.
 - Los dos últimos caracteres.
 - La cadena ingresada saltando de dos en dos caracteres. Por ejemplo, para la cadena ingresada 'bienvenido', debería imprimir 'bevnd'
 - La cadena en orden inverso. Por ejemplo, si se ingresa 'computadora', debe imprimir 'arodatupmoc'."

2. En un tranquilo día en la biblioteca, Sherlock Holmes, el famoso detective, se encuentra resolviendo acertijos literarios. Al revisar un manuscrito misterioso, nota que el texto contiene pistas ocultas. Implementa un programa que, al recibir un mensaje codificado, permita al intrépido detective contar el número de espacios en blanco en ese texto.
3. En una aventura literaria, Don Quijote, el valiente caballero, se sumerge en la biblioteca del castillo en busca de enigmas. Ayuda a Don Quijote a descifrar un antiguo manuscrito. Implementa un programa que, al recibir un mensaje encriptado, permita al caballero contar la cantidad de letras mayúsculas que contiene ese texto.
4. En la lejana tierra de Oz, Dorothy descubre un libro antiguo lleno de encantamientos. Para descifrar los hechizos, necesita contar los números mágicos ocultos en ellos. Implementa un programa que, al recibir un conjuro misterioso, permita a Dorothy contar la cantidad de dígitos que contiene. Si no hay ningún número, mostrará el mensaje "¡Este encantamiento no contiene ningún número!"
5. En las colinas encantadas de Avalon, el mago Merlín se enfrenta a un antiguo pergamino lleno de encantamientos. Para desvelar su misterio, necesita manipular el texto con distintas convenciones. Implementa un programa que, al recibir un hechizo secreto (puede combinar mayúsculas y minúsculas como quiera), permita a Merlín mostrar el hechizo tres veces: una vez con todas las letras minúsculas, otra con todas las letras mayúsculas y la última vez con la primera letra de cada palabra en mayúscula.
6. En los bosques del reino de Narnia, la Reina de los Hechizos necesita comprender los secretos de un antiguo grimorio. Implementa un programa que, al recibir un conjuro, permita a la Reina contar la cantidad de letras en mayúsculas y minúsculas presentes en él, revelando así los secretos codificados en la mezcla de letras
7. En el laberinto de la fortaleza encantada, el valiente caballero Sir Galahad necesita determinar cuántas veces se encuentran ciertos símbolos mágicos en los pasillos. Implementa un programa que, al recibir una cadena de caracteres y un carácter específico, cuente las veces que este último aparece en la cadena, ayudando así al caballero a desentrañar los secretos de este lugar misterioso. (No emplear la función count de Python)
8. Elabora un programa que ingresando una palabra por teclado muestre la palabra, sustituyendo las vocales por números, de la siguiente manera: a = 1; e = 2; i = 3; o = 4; u = 5.

9. En la lejana aldea de Camelot, el rey Arturo desea saber cuántas letras hay en el nombre de sus valientes caballeros. Implementa un programa que, al ingresar el nombre del caballero por consola, muestre como salida la cantidad de letras en su nombre.
10. En la moderna ciudad de Silicon Valley, una empresa de tecnología solicita ayuda para organizar su directorio telefónico. Implementa un programa que, al recibir un número de teléfono con formato +51 XXXXXXXXX XXX, muestre en pantalla únicamente el número central, omitiendo el prefijo del código del país y la extensión, para facilitar su registro en la guía telefónica de la compañía

Semana 10: Sesión 2

Listas - Tuplas

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 3

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándote de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El propósito de la siguiente actividad es desarrollar habilidades de creación, manipulación y acceso a elementos en listas y tuplas, lo que es fundamental para trabajar con datos estructurados.

II. Descripción de la actividad por realizar

1. Crea una lista con los nombres de 5 de tus amigos y luego imprime la lista completa. A partir de la lista creada, agrega un nuevo nombre al final de la lista e imprime la lista actualizada.
2. Dada la siguiente lista [2, 4, 8, 10, 12], implemente un programa que calcule la suma de los números de la lista.
3. Crea una tupla con los días de la semana e imprime la tupla completa. Luego completa el programa para que imprima el tercer día de la semana
4. Dada la siguiente lista [1, 2, 5, 2, 1, 6, 3, 4, 5, 6], implemente un programa que elimine los números duplicados de la lista.

5. Crea una lista de tuplas, donde cada tupla contiene un número y su cuadrado. La lista debe contener los números del 1 al 5 y sus cuadrados.
6. Dada la siguiente lista ["Ana", "Bob", "Cara", "Dana", "Eva"], escribe un programa que organice estos nombres en orden alfabético.
7. En una competencia escolar, se tienen registros de las puntuaciones obtenidas por los estudiantes en diferentes pruebas. Implementa un programa que, dada una lista de tuplas con los nombres de los estudiantes y sus respectivas calificaciones: [("Ana", 15), ("Bob", 10), ("Cara", 11)], encuentre al estudiante que obtuvo la nota más alta en las pruebas.
8. Eres el encargado del inventario de una tienda de moda. Tu tarea es mantener un registro de las prendas disponibles en la tienda. Cada prenda se registra con una tupla que contiene: nombre del artículo, tamaño (S, M, L), color y cantidad en existencia. Debe elaborar una lista que almacene las tuplas de las prendas disponibles, así como también, implementar un programa que permita agregar nuevas prendas a la lista, actualizar la cantidad en existencia de una prenda, así como también buscar y mostrar todas las prendas de un color específico
9. Como gerente de una aerolínea pequeña, tu labor implica rastrear diariamente los vuelos. Cada vuelo se registra a través de una tupla que contiene información clave: el código del vuelo, la ciudad de origen, la ciudad de destino, la hora de salida y la hora de llegada. Tu tarea es crear una lista que almacene estas tuplas para los vuelos programados. Además, debes implementar un programa que permita agregar nuevos vuelos a esta lista, buscar vuelos a través de su código para mostrar sus detalles y listar todos los vuelos programados desde una ciudad de origen específica

Semana 11: Sesión 2

Diccionarios

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 3

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándose de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El propósito de esta actividad es desarrollar habilidades de creación, manipulación y acceso a elementos en diccionarios, lo que es fundamental para trabajar con datos estructurados.

II. Descripción de la actividad por realizar

1. Elabora un diccionario llamado "frutas", con tres pares de valores clave. Usa los nombres de las frutas como claves y la cantidad que posees de cada una como valores.
2. Dado el siguiente diccionario: colores = {"rojo": "red", "verde": "green", "azul": "blue"} Escribe un programa que muestre como salida la traducción al inglés del color "verde".
3. Dado el siguiente diccionario: animales = {"gato": 4, "perro": 4, "elefante": 4} Agrega un par de valores clave al diccionario para representar un animal y la cantidad de patas que tiene. Elige cualquier animal que prefieras.
4. Dado el siguiente diccionario: estudiantes = {"Ana": 22, "Luis": 21, "Pedro": 23} Implementa un programa que verifique si "Luis" está en el diccionario.

5. Dado dos diccionarios, por ejemplo: `dic1 = {"a": 1, "b": 2}` y `dic2 = {"b": 3, "c": 4}` Escribe un programa que fusione los dos diccionarios en uno. Si hay claves que aparecen en ambos diccionarios, elige el valor del segundo diccionario.
6. Implementa un programa que reciba una lista de números, por ejemplo, `"numeros = [1,2,2,3,3,3,4,4,4,4]"`, y genere un diccionario. Las claves de este diccionario deben ser los números de la lista, y los valores deben representar la frecuencia de aparición de cada número.
7. Dado un diccionario, por ejemplo: `personas = {"Pedro": 23, "Ana": 21, "Javier": 25, "Laura": 22}` Escribe un programa que busque una clave en el diccionario y devuelva el valor si la clave existe, en caso contrario, devuelva un mensaje indicando que la clave no se encuentra en el diccionario.
8. Dado un diccionario, por ejemplo: `productos = {"manzanas": 5, "plátanos": 3, "uvas": 4}`. Implementa un programa que ordene e imprima el diccionario según sus valores en orden ascendente.
9. Se requiere optimizar el sistema de gestión de inventario de una tienda minorista. Actualmente, cuentan con una lista de productos y sus cantidades en existencia. El objetivo es mejorar este sistema empleando diccionarios. Se debe crear un diccionario donde las claves representen los nombres de los productos y los valores reflejen las cantidades disponibles. El programa debe permitir agregar un nuevo producto al inventario, actualizar la cantidad de un producto existente, eliminar un producto del inventario, buscar y mostrar la cantidad disponible de un producto específico.
10. Eres el encargado del sistema de calificaciones de estudiantes de una universidad. Actualmente, el sistema usa listas para almacenar las calificaciones. Tu tarea es mejorar este sistema utilizando diccionarios. Debes crear un diccionario donde las claves representen los nombres de los estudiantes y los valores, sus calificaciones. Implementa un programa que permita agregar las calificaciones de un nuevo estudiante, actualizarlas, eliminarlas, buscar y mostrar las calificaciones de un estudiante específico.

Semana 12: Sesión 2

Estructuras de datos básicas

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 3

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándose de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El propósito de esta actividad es desarrollar habilidades para diferenciar y manipular estructuras de datos básicas aplicándolas correctamente según los requerimientos del problema. Esto les permitirá desarrollar programas más eficientes y efectivos.

II. Descripción de la actividad por realizar

1. El sistema actual de gestión de biblioteca de una universidad que registra los libros disponibles, los préstamos, las devoluciones y los perfiles de los usuarios, está anticuado y presenta problemas de rendimiento. Se les ha asignado la tarea de rediseñar un nuevo sistema. El proyecto consta de varias partes, y la primera tarea es implementar una forma eficiente de almacenar y manipular la información de los libros en la biblioteca. Cada libro tiene un título, un autor, una categoría (por ejemplo, ciencia, arte, literatura), una identificación única (ISBN) y un estado (disponible o en préstamo). Implementar un programa que permita buscar un libro por título o autor, verificar si un libro está disponible o en préstamo, cambiar el estado de un libro cuando se presta o se devuelve, mostrar todos los libros en una categoría específica.

2. El sistema actual de gestión de RRHH de una gran corporación que tiene múltiples departamentos y cientos de empleados, utiliza múltiples hojas de cálculo que son difíciles de mantener y propensas a errores. Es difícil buscar información específica o realizar cambios en los datos de los empleados como datos personales, roles, salarios y días de vacaciones. Tu tarea es diseñar e implementar utilizando estructuras de datos básicas, un mejor sistema de seguimiento, que permita almacenar y manipular eficiente de la información de los empleados.

Cuarta **Unidad**

Funciones y modularidad

Semana 13: Sesión 2

Funciones

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 4

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándote de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El propósito de esta actividad es desarrollar habilidades para definir funciones; llamarlas, manejar parámetros y argumentos, y valorar su importancia en la programación modular y la reutilización de código, lo que es esencial para abordar proyectos más grandes y complejos.

II. Descripción de la actividad por realizar

1. Implementa una función denominada "suma" que tome dos números como argumentos y devuelva su suma. Por ejemplo, "suma(3, 5)" debería devolver "8".
2. Implementa una función denominada "mayor" que tiene como argumento una lista de números y devuelva el mayor número de la lista.
3. Implementa una función denominada "factorial" que tome un número entero positivo como argumento y devuelva su factorial.
4. Implementa una función denominada "generar_potencias(base, n)" que genere una lista con las "n" primeras potencias de "base". Por ejemplo, "generar_potencias(2, 5)" debería devolver "[2, 4, 8, 16, 32]".

5. Implementa una función denominada "Fibonacci" que tome un número "n" como argumento y devuelva los primeros "n" números de la serie Fibonacci. La serie Fibonacci es: 0, 1, 1, 2, 3, 5, 8, 13, ... Por ejemplo, "Fibonacci(7)" debería devolver "[0, 1, 1, 2, 3, 5, 8]".
6. Implementa una función denominada "filtrar_palabras(lista, n)" que tome como argumentos una lista de palabras y un entero "n", y retorne una lista con las palabras que tengan más de "n" caracteres. Por ejemplo, "filtrar_palabras(["manzana", "banana", "kiwi"], 5)" debería devolver '["manzana", "banana"]'
7. Eres un desarrollador de software que trabaja en un sistema de gestión de gastos para una empresa. Necesitas implementar un programa que calcule el total de los gastos de una lista dada. Cada gasto se registra como un diccionario con información sobre la fecha, la descripción del gasto y el monto. Tu objetivo es desarrollar una función que reciba como argumento una lista de gastos y devuelva la suma total de todos los gastos.
8. Estás colaborando en el desarrollo de un sistema de evaluación para una plataforma de aprendizaje en línea. Necesitas implementar un programa que calcule el promedio ponderado de las calificaciones de un estudiante basándose en una lista de calificaciones proporcionada. Cada calificación está registrada como un diccionario con información sobre el nombre de la asignatura, la calificación y el número de créditos de esa asignatura. Tu tarea es implementar una función que tome como argumento una lista de calificaciones y devuelva el promedio ponderado calculado. Asegúrate de considerar los diferentes pesos de los créditos en el cálculo del promedio ponderado

Semana 14: Sesión 2

Modularidad

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 4

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándote de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El propósito de esta actividad es desarrollar habilidades necesarias para implementar código modular, reutilizable y escalable; valorando la importancia de la programación modular, que es esencial para abordar proyectos más grandes y complejos.

II. Descripción de la actividad por realizar

1. Implementa un archivo denominado *matematicas.py*, el cual debe contener una función denominada *suma* que acepte dos argumentos y devuelva la suma de los números. En otro archivo Python, importa el módulo *matematicas* y usa la función *suma* para sumar dos números y mostrar el resultado.
2. Implementa un archivo denominado *circulo.py*, el cual debe contener una función denominada *area_del_circulo* que acepte el radio como argumento y devuelva el área del círculo. En otro archivo Python, importa el módulo *circulo* y utiliza la función *area_del_circulo* para determinar el área de un círculo con un radio específico y muestra el resultado.

3. Implementa un archivo denominado *conversion.py*, el cual debe contener dos funciones: *metros_a_pies* que convierte metros a pies y *pies_a_metros* que convierte pies a metros. En otro archivo Python, importa el módulo *conversion* y utiliza ambas funciones para convertir una longitud dada entre metros y pies.
4. Implementa un archivo denominado *numeros.py*, el cual debe contener una lista de números. En otro archivo Python, importa el módulo *numeros* y utiliza la lista para realizar operaciones como encontrar el número máximo, mínimo o calcular la suma de los números.
5. Implementa un archivo denominado *saludos.py*, en el cual debe contener una función denominada *saludar* que acepte un nombre como argumento y devuelva un saludo personalizado. En otro archivo Python, importa el módulo *saludos* y utiliza la función *saludar* para saludar a diferentes personas llamando a la función con diferentes nombres.
6. Implementa un archivo denominado *estadisticas.py*, el cual debe contener funciones para determinar la media, mediana y desviación estándar de una lista de números. En otro archivo Python, importa el módulo *estadisticas* y utiliza estas funciones para realizar cálculos estadísticos en un conjunto de datos.
7. Como desarrollador en un pequeño consultorio médico, necesitas crear un sistema para registrar a los pacientes y sus citas. Se requiere que:
 - Dentro de un módulo "consultorio", crea una función "registrar_paciente" que tome como argumentos el nombre y la edad del paciente y devuelva un diccionario con esos datos.
 - Crea una función "agendar_cita" que tome como argumentos un diccionario de paciente y una fecha de cita (como una cadena) y agregue la fecha de la cita al diccionario del paciente.
 - Realiza una prueba de tus funciones creando un nuevo paciente, agendando una cita y luego imprimiendo los datos del paciente.
8. Estás desarrollando el sistema de administración de una tienda y necesitas crear un módulo para manejar el stock de productos. Se requiere que:
 - Dentro de un módulo "tienda", crea una función "agregar_producto" que tome un nombre de producto y una

cantidad, y almacene estos datos en un diccionario (el nombre del producto como la clave y la cantidad como el valor).

- Crea una función "vender_producto" que tome el nombre de un producto y una cantidad, y actualice la cantidad de ese producto en el diccionario de stock.
 - Asegúrate de que la función "vender_producto" maneje correctamente los casos en los que intentas vender más productos de los que hay en stock.
 - Realiza una prueba de tus funciones agregando algunos productos, vendiendo algunos de ellos e imprimiendo el stock de productos después de cada operación.
9. Un hotel local está desarrollando un nuevo sistema de reservas para gestionar sus habitaciones. El sistema debe ser capaz de gestionar diferentes tipos de habitaciones (individual, doble, suite, etc.) con diferentes precios. Además, el hotel ofrece descuentos para las reservaciones de más de 3 días. Se requiere que:
- Diseñe un módulo que contenga las funciones necesarias para gestionar las reservaciones. Esto puede incluir funciones para añadir una nueva reserva, calcular el costo de una reserva, aplicar descuentos, etc.
 - Implemente la funcionalidad para calcular el costo de una reserva, considerando el tipo de habitación y la duración de la estancia. Asegúrese de aplicar correctamente el descuento ofrecido por el hotel.
 - Su código debe estar diseñado de tal manera que pueda ser fácilmente extendido en el futuro, por ejemplo, para añadir nuevos tipos de habitaciones o cambiar las reglas de descuento.

Semana 15: Sesión 2

Recursividad

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 4

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándose de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El propósito de esta actividad es fortalecer las habilidades en resolución de problemas, diseño de algoritmos y programación eficiente, haciendo uso de la recursividad. Esta técnica, versátil y poderosa, se emplea en una amplia gama de aplicaciones, y busca potenciar la capacidad para abordar desafíos complejos con enfoques más flexibles y eficaces.

II. Descripción de la actividad por realizar

1. Implementa una función recursiva en Python que determine la suma de los "n" primeros números naturales, emplear:

$$\sum_{i=1}^n i = \begin{cases} 1, & \text{si } n = 1; \\ n + \sum_{i=1}^{n-1} i, & \text{si } n > 1. \end{cases}$$

2. El profesor Pitágoras necesita de tu ayuda para resolver un problema matemático. Implementa una función recursiva que determine el número de dígitos presentes en un número entero dado.

3. En una empresa de análisis de datos, se requiere calcular el factorial de números para realizar algunos cálculos estadísticos. Te asignan para que elabores una función recursiva denominada "calcular_factorial" que tome como argumento un número y retorne el factorial de ese número.
4. Implementa una función recursiva denominada "suma_digitos" que tome como argumento un número "n" y retorne la suma de los dígitos de "n".
5. Implementa una función recursiva denominada "suma_lista", la cual reciba como parámetro una lista de números y devuelva la suma total de sus elementos. En caso de que la lista esté vacía, la función deberá retornar 0."
6. Implementa una función recursiva denominada "invertir_cadena". Esta función deberá recibir una cadena como parámetro y devolver dicha cadena, pero invertida
7. Implementa una función recursiva denominada "contar_ocurrencias". Esta función debe recibir una cadena y un carácter como argumentos, y retornar el número de veces que ese carácter aparece en la cadena. Cuando la cadena esté vacía, la función devolverá 0
8. Implementa una función recursiva denominada "potencia" que tome como argumentos un número "n" y un número "p", y retorne "n" elevado a la potencia "p".
9. Una empresa necesita gestionar su inventario de productos. La empresa tiene diferentes tipos de productos y cada uno tiene su propio stock. Te han encargado que elabores una función recursiva denominada "calcular_inventario" que tome como argumento una lista de productos, donde cada producto es un diccionario con dos campos: "nombre" y "cantidad", esta debe retornar la suma total de todos los productos en el inventario. Cuando la lista de productos esté vacía, la función debe retornar 0.

Semana 16: Sesión 2

Programación modular

Sección: Fecha:/...../..... Duración: 60 minutos

Docente: Unidad: 4

Nombres y apellidos:

Instrucciones

Lea detenidamente cada enunciado asegurándote de comprender claramente cuál es el problema que se debe resolver y los requisitos específicos o restricciones que debes cumplir.

I. Propósito

El propósito de esta actividad es desarrollar habilidades para aplicar la programación modular y el uso de librerías en la resolución de problemas.

III. Descripción de la actividad por realizar

1. La empresa de tecnología "TechInnov" para la que trabajas, ha sido contratada por una gran cadena de supermercados, SuperMart, para desarrollar un nuevo sistema de gestión de inventario.

SuperMart tiene una gran variedad de productos en sus tiendas, y ha estado teniendo problemas para seguir la pista de todos ellos. El antiguo sistema de gestión de inventario es obsoleto y a menudo conduce a errores, lo que resulta en pedidos incorrectos, pérdida de ventas y clientes insatisfechos.

SuperMart necesita un sistema de gestión de inventario más eficiente y confiable que pueda manejar la cantidad y variedad de productos en sus tiendas. Tu empresa te ha asignado el trabajo para que implementes el sistema que debe ser capaz de realizar varias tareas, como agregar productos, restar productos cuando se venden, revisar el inventario y reordenar productos cuando se bajan de cierto nivel.

2. La empresa de tecnología "ProgSolutions" para la que trabajas, ha sido contratada por constructora "BuildRise" para desarrollar una aplicación de gestión de proyectos.

"BuildRise" maneja múltiples proyectos de construcción simultáneamente. Cada proyecto tiene varios equipos de trabajadores, cada uno con diferentes roles, tareas y plazos para sus tareas. Han tenido dificultades para coordinar eficazmente los equipos y las tareas, lo que ha llevado a retrasos y aumento de costos.

La aplicación que tu equipo debe desarrollar debe permitir a los gerentes de proyecto, crear proyectos, agregar equipos a los proyectos, asignar roles y tareas a los miembros del equipo, establecer plazos para las tareas, y monitorear el progreso de las tareas.

Referencias

Chazallet, S. (2020). *Python 3 - Los fundamentos del lenguaje*. (3ª ed.) Ediciones ENI.

Cóndor E., De La Cruz M. (2020). *Algoritmos resueltos con Python*. Editorial EIDEC.

Ebel, F., Rohaut, S. (2019). *Algoritmia - Técnicas fundamentales de programación: Ejemplos en Python*. Ediciones ENI.

Joyanes, L. (2020). *Fundamentos de programación: algoritmos, estructura de datos y objetos*. (5º ed.) McGraw-Hill.

Marzal A., Gracia i., García P. (2014). *Introducción a la programación con Python 3*. España: Editorial Publicacions de la Universitat Jaume I.

Microsoft Corporation. (2023). Visual Studio Code [Software]. <https://code.visualstudio.com/>

Python Software Foundation. (2023). Python. [Software]. <https://www.python.org/downloads/>

Reinoso I. *Diccionarios Python: Guía + Ejercicios 2023*. BigBayData.com. Recuperado el 31/10/2023 de <https://www.bigbaydata.com/diccionarios-python/>

Sánchez A. (11 de febrero de 2021). *Ejercicios de Cadenas*. Aprende con Alf. Recuperado el 02/11/2023 de <https://aprendeconalf.es/docencia/python/ejercicios/cadenas/>

Sintes B. (6 de octubre de 2023) *Introducción a la programación con Python*. mclibre. <https://www.mclibre.org/consultar/python/>