

Universidad CENFOTEC



Escuela de Software

Código del curso: SOFT-01.

Nombre del curso: Principios de Programación 1.

Sección: SCV3.

Periodo: C3-2025.

Docente facilitador: Stephanie Delgado Brenes.

CONSIGNA DEL SEGUNDO AVANCE DEL PROGRAMA DE COMPUTADORA

1. Datos generales de la actividad

Tipo de actividad: Avance de proyecto.

Fecha de entrega: 09 de noviembre del 2025, 23:59pm. **Valor porcentual:** 15%.

Formato de entrega: Archivo comprimido. **Puntaje total:** 40.

Individual: No. **Grupal:** Sí.

2. Instrucciones generales

1. Lea cuidadosamente las instrucciones de la actividad. En caso de tener alguna duda, puede consultar con su docente.
2. Esta actividad se desarrolla de manera grupal. Cualquier intento de plagio será sancionado de acuerdo con el reglamento académico vigente.
3. Al completar la actividad, un representante del grupo debe subir un archivo comprimido a la plataforma Moodle en el tiempo y espacio indicado por el docente. El archivo comprimido debe contener un archivo **.pdf** con los diagramas y la documentación solicitada y los archivos de código fuente **.py** requeridos para la implementación completa del programa descrito a continuación.

3. Objetivos o competencias del curso que se evaluarán en la actividad de aprendizaje

Objetivo general o competencia del curso	<p>Desarrollar el pensamiento lógico-computacional para la resolución de problemas bien definidos de complejidad baja, utilizando el paradigma imperativo, un lenguaje de alto nivel y un razonamiento riguroso de forma precisa.</p>
Objetivos específicos que se evalúan	<ul style="list-style-type: none"> • Implementar un programa de computadora en el paradigma imperativo, usando algoritmos, estructuras computacionales, criterios de calidad y buenas prácticas de programación. • Elaborar un documento que incluye los diagramas del problema y las tablas de datos: de entrada, de salida, intermedios y constantes. • Confeccionar el diagrama general, el diagrama explicativo, el diagrama de flujo final y el diseño de los casos de prueba. • Realizar un programa de computadora en un lenguaje de programación de alto nivel, de acuerdo con los criterios y estructuras vistas en el curso. • Aplicar las diferentes estructuras (secuenciales, condicionales, iterativas, entre otras) en la implementación computacional. • Verificar la funcionalidad del programa utilizando como mínimo los dos casos de prueba diseñados. • Utilizar buenas prácticas en la codificación del programa: robustez, nitidez del programa, el seguimiento a estándares, entre otros.

4. Descripción de la actividad

Contexto:

El programa de computadora promueve la aplicación de los fundamentos de programación estructurada, imperativa y procedimental necesarios para la construcción de algoritmos y la resolución de problemas computacionales mediante su análisis y el diseño e implementación de programas de computadora.

Instrucciones:

El grupo de trabajo debe implementar los siguientes programas en Python. Los programas deberán ser las opciones de un menú de consola simple, en el cual las opciones de la **1** a la **4** serán las opciones válidas correspondientes con los programas solicitados, **0** será la opción para salir, y cualquier otro dato ingresado por el usuario que no sea un número entero con valor de **0**, **1**, **2**, **3** ó **4** será considerado un dato inválido, por lo que deberá solicitarse de nuevo el ingreso de la opción.

1. Una optimización del ejercicio 1 del primer avance del programa de computadora, en el que se valide que el número recibido del usuario es del tipo adecuado (i.e., entero) y está dentro del rango válido (i.e., de **0** a **20**); mientras no se haya recibido un dato que cumpla con las exigencias, el programa dará un mensaje amigable con el usuario para indicar que se requiere el ingreso de otro número que sí las cumpla. Una vez que se haya verificado la recepción de un número adecuado, el programa realizará el análisis indicado por las instrucciones de la versión anterior del ejercicio y comunicará si el juego se encuentra en curso o si el jugador ha perdido la partida, según sea el caso.

2. Una optimización del ejercicio 2 del primer avance del programa de computadora, en el que se valide que el número recibido del usuario es del tipo adecuado (i.e., entero) y está dentro del rango válido (i.e., de **0** a **50**); mientras no se haya recibido un dato que cumpla con las exigencias, el programa dará un mensaje amigable con el usuario para indicar que se requiere el ingreso de otro número que sí las cumpla. Una vez que se haya verificado la recepción de un número adecuado, el programa realizará el análisis indicado por las instrucciones de la versión anterior del ejercicio y comunicará la cantidad de puntos acumulada por el jugador durante la partida.

3. Una optimización del ejercicio 3 del primer avance del programa de computadora, en el que se valide que el carácter recibido del usuario es del tipo adecuado (i.e., **str**) y está dentro de la colección de letras válidas (i.e., '**o**', '**l**', '**s**', '**z**', '**L**', '**J**' ó '**T**'); mientras no se haya recibido un dato que cumpla con las exigencias, el programa dará un mensaje amigable con el usuario para indicar que se requiere el ingreso de otro carácter que sí las cumpla. Una vez que se haya verificado la recepción de un carácter adecuado, el programa realizará el análisis indicado por las instrucciones de la versión anterior del ejercicio e imprimirá en la consola la correspondiente figura mediante asteriscos.

4. Un programa que reciba del usuario un carácter y dos números enteros positivos: el carácter representando la orientación de una pieza (i.e., '**A**' para arriba, '**a**' para abajo, '**d**' para derecha e '**i**' para izquierda), y los números representando la cantidad de rotaciones a la derecha y a la izquierda respectivamente. Una vez que se

haya validado que los datos recibidos tienen tipos y valores adecuados – solicitándolos de nuevo las veces que sea necesario, tras la impresión de un mensaje explicativo amigable con el usuario – el programa determinará e imprimirá en pantalla la orientación final de la pieza. **Ejemplo:** Si se ingresa el carácter ‘d’ y los nuevos **7** y **4**, quiere decir que la pieza inicia con orientación a la derecha y recibe **7** rotaciones derechas y **4** rotaciones izquierdas, lo cual equivale a **3** rotaciones derechas netas, de modo que la impresión final debería indicar que la orientación final de la pieza es ‘A’.

5. Rúbrica

Esta actividad de aprendizaje será evaluada mediante la siguiente rúbrica:

Criterio de evaluación	Insuficiente (1 punto)	Aceptable (2 puntos)	Bueno (3.5 puntos)	Excelente (5 puntos)
Diagramación	Elaboración inadecuada o incompleta de los diagramas generales, explicativos o de flujo para los diferentes programas.	Elaboración superficial de los diagramas generales, explicativos o de flujo para los diferentes programas.	Elaboración básica de los diagramas generales, explicativos o de flujo para los diferentes programas.	Elaboración correcta y exhaustiva de los diagramas generales, explicativos o de flujo para los diferentes programas.
Determinación de datos del programa	Determinación inadecuada o incompleta de los datos del programa en tablas de variables de entrada, de salida, intermedias y constantes de cada programa.	Determinación parcial de los datos del programa en tablas de variables de entrada, de salida, intermedias y constantes de cada programa.	Determinación básica de los datos del programa en tablas de variables de entrada, de salida, intermedias y constantes de cada programa.	Determinación completa y ejemplificada de los datos del programa en tablas de variables de entrada, de salida, intermedias y constantes de cada programa.
Documentación de casos de prueba	Documentación incompleta o confusa del funcionamiento de cada programa.	Documentación superficial del funcionamiento de cada programa.	Documentación clara pero básica del funcionamiento de cada programa.	Documentación completa y exhaustiva del funcionamiento de cada programa.

Aplicación de buenas prácticas y estándares de programación	Buenas prácticas y estándares de programación aplicados pocas veces en cada programa.	Buenas prácticas y estándares de programación aplicados frecuentemente en cada programa.	Buenas prácticas y estándares de programación aplicados en la mayoría de cada programa.	Cumplimiento de las buenas prácticas y estándares de programación en la totalidad de cada programa.
Aplicación de lógica de programación	Lógica inadecuada de programación en cada programa.	Lógica parcialmente adecuada de programación en cada programa.	Lógica mayoritariamente adecuada de programación en cada programa.	Lógica robusta y adecuada de programación en cada programa.
Validación de los datos de entrada	Los datos de entrada de los programas no se validan, o se hace con errores graves.	La validación de los datos de entrada es superficial o tiene errores considerables.	La validación de los datos es efectiva y correcta en su gran mayoría.	La validación de los datos se lleva a cabo de forma completa y correcta, sin ningún error.
Implementación del menú	No se presenta menú, o el menú no es funcional	Menú funcional pero con considerables errores	Menú funcional pero poco intuitivo o amigable con el usuario	Menú plenamente funcional, intuitivo y amigable con el usuario, libre de errores.
Cumplimiento de requisitos	Cumple una cuarta parte o menos de los requisitos solicitados.	Cumple al menos la mitad de los requisitos solicitados.	Cumple tres cuartas partes de los requisitos solicitados.	Cumple la totalidad de los requisitos solicitados.