

**ASIGNATURA:**

**Fundamentos de Programación**



**Evaluación Final- Examen**

# APRENDIZAJE ESPERADO

En esta podrás trabajar todo aquello que has realizado durante la asignatura considerando diagramas de flujo de datos, variables, operadores, instrucciones de control, arreglos (listas) y funciones manera que puedas implementar soluciones integrales y efectivas dentro de un programa computacional.

## INSTRUCCIONES GENERALES

## Estimado(a) estudiante:

* Una vez realizada la lectura comprensiva del material de estudio de la asignatura, los ejercicios y recursos de aprendizaje dispuestos en el aula, te invitamos a desarrollar la evaluación final del curso.
* Para responder las preguntas y/o realizar la siguiente actividad, refiérase principalmente a los contenidos trabajados en las semanas previas. También puedes recurrir a fuentes de información segura en Internet. En este caso debes declarar las referencias.



**Esta actividad se complementa con una pauta de autoevaluación al finalizar, para que verifiques el logro**

**de tus aprendizajes**

# EJERCICIO 1 – 20 puntos

Genere un Diagrama de Flujo de Datos (DFD) que refleje el proceso para imprimir todos los números entre 1 y 1.000.000 que cumplen con la condición de ser par y, al mismo tiempo, ser divisibles (múltiplos) de 3.

|  |
| --- |

# EJERCICIO 2 – 40 puntos

El medidor de lluvia es un sistema que permite recoger información sobre los mm de agua que han caído en Santiago. Una muestra tendrá N mediciones, donde cada medición corresponde a una cantidad de mm de agua caída, por lo que se pide que genere un programa en Python 3 capaz de:

* Solicitar la cantidad de mediciones a registrar para la muestra y guardar dicho número en la variable “N”. Se debe asegurar que “N” siempre sea un número entero positivo (en caso de que se ingrese cero o un numero negativo deberá solicitar, tantas veces como sea necesario, nuevamente el valor para “N” hasta que se cumpla lo requerido).
* Solicitar y validar cada medición ingresada de modo que estén en el intervalo de 0 a 150 (ambos inclusive). La validación es medición a medición. Es decir, cada vez que se ingrese una medición no válida (fuera del intervalo señalado) deberá enviar un mensaje que diga “Error, ingrese una medida en el intervalo de 0 a 150” y, luego, dar oportunidad de corregir dicha medida. Esto deberá ser realizado tantas veces como sea necesario.

Una vez ingresadas las N mediciones válidas el programa imprimirá el mensaje: “El promedio de agua caída es: <AGUA>” donde <AGUA> representa el valor del promedio simple de todas las mediciones válidas que fueron ingresadas.

***IMPORTANTE: La entrega de este ejercicio contempla un único archivo en formato .py (un solo programa de Python con todo el código para el enunciado solicitado)***

# EJERCICIO 3 – 40 puntos

La antigua ruleta de casino tenía los números del 0 al 36 y a los jugadores les costaba mucho realizar cálculos matemáticos para apostar con las mayores probabilidades. Suponiendo que tenemos una lista llamada “ruleta” de largo 37 y que en cada “casilla” (índice) se almacena la cantidad de veces que ese número salió sorteado (ej. Si en la casilla 25 hay un 12 significa que el número 25 ha salido 12 veces), se pide que escriba las siguientes funciones:

1. **noHanSalido(ruleta)** que recibe la lista que representa la ruleta e imprime por pantalla los números que no han sido sorteados (que no han salido).
2. **actualizarRuleta(ruleta, N)** que recibe la lista que representa la ruleta y un número entero “N” y luego suma 1 a la cantidad de veces que ha sido sorteado un número “N”. Debe validar que el valor de “N” sea un número entre 0 y 36 (ambos inclusive). Si el número no corresponde al rango estipulado deberá imprimir el valor –1.
3. **obtenerPorcentaje(ruleta, N)** que recibe la lista que representa la ruleta y un número entero “N” y retorna el porcentaje (número decimal) de veces que ha sido sorteado un número “N” sobre el total de veces que se ha tirado la ruleta. Asuma que, en este caso, el calor de “N” siempre será número entre 0 y 36 (ambos inclusive) y, por ende, no lo necesita validar.

***IMPORTANTE: La entrega de este ejercicio contempla un único archivo en formato .py (un solo programa de Python con todo el código para el enunciado solicitado)***

## PAUTA DE EVALUACIÓN

| **Criterios de Evaluación** | **Indicadores de Logro** | **Puntaje** |
| --- | --- | --- |
| El diagrama de flujo es generado en concordancia con los requerimientos y contiene secuencias, decisiones e iteraciones del algoritmo solicitado. | Ejercicio 1: Marca el inicio y el Fin del Diagrama | 02 |
| Ejercicio 1: Se definen las variables del diagrama correctamente | 03 |
| Ejercicio 1: Se define la operación para determinar si un número es par | 03 |
| Ejercicio 1: Se define la operación para determinar si un número es divisible (múltiplo) de tres | 03 |
| Ejercicio 1: Se genera el flujo para la diferenciación de los números que simultáneamente son múltiplos de 3 y par | 03 |
| Ejercicio 1: El diagrama genera el flujo correcto para imprimir los valores solicitados. | 06 |
| Construir algoritmos en lenguaje de programación que utilicen uno o más de los siguientes componentes: variables y tipos de datos fundamentales, IF, FOR, WHILE, Funciones y/o Arreglos (listas) en concordancia con los requerimientos del ejercicio planteado. | Ejercicio 2: Realiza correctamente las operaciones de input para garantizar el valor de N | 10 |
| Ejercicio 2: Realiza correctamente las operaciones de input para garantizar los valores de cada medición | 10 |
| Ejercicio 2: Genera un procedimiento correcto para acumular o utiliza una lista para almacenar los datos de todas las temperaturas válidamente ingresadas. | 10 |
| Ejercicio 2: Obtiene el promedio correcto de las temperaturas | 10 |
| Ejercicio 3: Declara correctamente la función del punto A) considerando los parámetros estipulados | 02 |
| Ejercicio 3: Declara correctamente la función del punto B) considerando los parámetros estipulados | 02 |
| Ejercicio 3: Declara correctamente la función del punto C) considerando los parámetros estipulados | 02 |
| Genera el procedimiento interno de la función solicitada en la letra A) de manera de imprimir los números que “nunca han salido” | 08 |
| Genera el procedimiento interno de la función solicitada en la letra B) de manera de mostrar la cantidad de veces que ha salido el número “N” | 08 |
| Realiza la validación del valor de “N” de manera de imprimir la cantidad de veces que ha sido sorteado o un –1 en su defecto. | 05 |
| Genera el procedimiento interno de la función solicitada en la letra C) de manera de mostrar el porcentaje (decimal) la cantidad de veces que ha salido el número “N” sobre el total de veces que se ha tirado la ruleta | 08 |
| Realiza correctamente el cálculo o utiliza una función predefinida del lenguaje para obtener la cantidad de veces que se ha tirado la ruleta. | 05 |
| Puntaje Total | | 100 |

## PAUTA DE AUTOEVALUACIÓN:

A continuación, te invitamos a responder una pauta de autoevaluación, para que verifiques lo aprendido en esta actividad y el logro del o los objetivos planteados inicialmente. Recuerda, en caso de presentar dudas, debes participar de forma permanente en los foros, donde el docente te retroalimentará y dará respuesta a tus inquietudes.

| **Criterios de revisión del informe con la información recopilada:** | **Completamente logrado**  **(4)** | **Medianamente logrado**  **(3)** | **Parcialmente logrado**  **(2)** | **No logrado**  **(0)** |
| --- | --- | --- | --- | --- |
| 1. Pude idear la solución a los ejercicios planteados en virtud de los requerimientos exigidos | 4 |  |  |  |
| 1. Pude realizar las operaciones solicitadas en lo que respecta a instrucciones de control | 4 |  |  |  |
| 1. Logré identificar los parámetros solicitados en lo relativo a funciones | 4 |  |  |  |
| 1. Generé el o los retornos y resultados esperados en virtud de lo solicitado | 4 |  |  |  |
| 1. Generé el DFD solicitado considerando su estructura y componentes | 4 |  |  |  |
| **Puntaje Total** | **20 puntos** | | | |
| **Puntaje Obtenido** | **XX Puntos obtenidos** | | | |

## ESCALA DE NOTAS





