# **EVALUACION 4:**

# Sumativa 4: Programa básico en Python

NOMBRE: Javier Guarda Acosta

CARRERA: Analista Programador

ASIGNATURA: Introducción a Programación y Robótica Aplicada

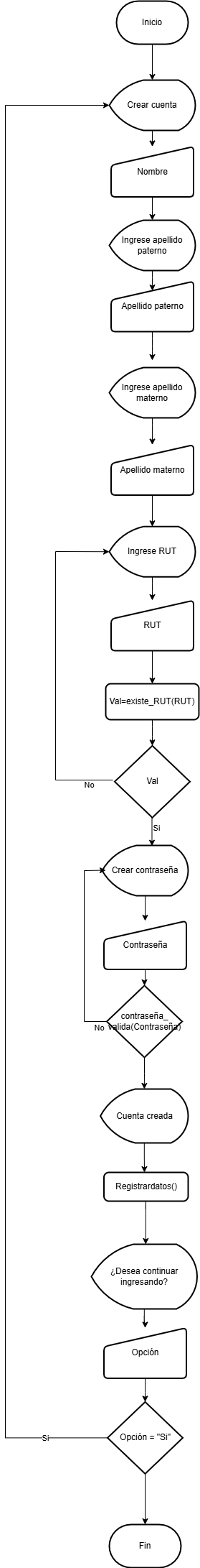
PROFESOR: Jacqueline San Martin

FECHA: 02-06-2025

# Introducción

En el presente trabajo se interpretará el diagrama de flujo otorgado en el inicio de la actividad que cumple la funcionalidad de registrar usuarios, consultando por su nombre, apellidos, RUT y contraseña. Al terminar las entradas de datos del usuario, se le consultará si desea registrar más usuarios al sistema, de ser así comienza el flujo de nuevo, si no, el programa acaba. Se deberán tomar decisiones como la estructura de datos a almacenar los inputs recibidos en memoria, y principalmente estructurar el flujo del programa en un script ordenado que represente al diagrama de flujo y se componga de bloques de condicionalidad y funciones.

El diagrama de flujo a representar es el siguiente:



# Desarrollo

Para desarrollar el script se eligió el IDE Visual Studio Code, ya que es la herramienta con la que los estudiantes estamos familiarizados y manejamos día a día, además de por la popularidad que posee.

Como lenguajes de programación se utilizará Python, básicamente por las mismas razones referidas anteriormente.

Se harán las siguientes validaciones:

* Todas las entradas para nombre completo (nombre, apellido paterno y apellido materno), no deberán extenderse a más de 15 letras. Además, en una misma sentencia condicional se evalúa que sólo se ingresen letras a cada input, enviándolos como argumento a una función que utiliza el método de Python isalpha(), que retorna *True* en caso de recibir solo caracteres alfanuméricos, o *False* en caso contrario.
* El RUT será validado enviando el input recibido mediante una función en la cual se evaluará con una expresión regular, la que lo validará sólo si posee entre 8 a 9 caracteres al tratarse de la región de Chile, además de que el penúltimo carácter deberá ser un guión, y seguido de este deberá haber un digito del 0 al 9 o la letra k. Se validará indistintamente si los dígitos antes del guión están o no separados por un punto, es decir, ambas formas serán válidas.
* La contraseña constará de las siguientes reglas: la contraseña debe tener mínimo 8 caracteres, máximo 15, al menos una letra mayúscula, al menos una letra minúscula, al menos un dígito, no espacios en blanco, al menos 1 caracter especial. Estos requisitos serán evaluados mediante la expresión regular correspondiente, luego de la cual, al validarse el formato de la contraseña, se le pedirá al usuario repetir la contraseña, donde en caso de ser iguales, el flujo avanzará a la función donde se envían los datos a una función de registro, y si no son iguales, se inicia nuevamente el ciclo *while True* donde se introduce la contraseña y se valida su formato.
* Al cumplirse todas las validaciones, se envían los valores guardados en inputs a la función de registro como argumentos de esta. En dicha función se almacenan los datos en el diccionario ‘usuarios’ que se inicializa como vacío y como variable global del programa. Aquí, el RUT del usuario funciona como clave de cada usuario, y como valor, una tupla que posee el nombre, apellido paterno, apellido materno y contraseña de cada usuario.

En cuanto a tiempo de ejecución, el programa tarda de ejecutarse aproximadamente 3 minutos de forma fluida al ingresarse un primer usuario, luego se le pregunta al usuario registrador si desea seguir ingresando personas, para continuar en el ciclo. Si no desea seguir ingresando, vuelve al Menú principal, donde puede seguir en el programa viendo las demás opciones que ofrece, o definitivamente salir de éste.

# Conclusiones

Como mejoras al algoritmo, se decide implementar un Menú Principal con opciones para el usuario, que aparecería al inicio del programa. Este menú tiene 5 opciones que dirigirán al programa a diferentes funciones dentro del script, las cuales son Registrar Usuarios (que conduce al algoritmo principal de Crear Cuenta, donde se representan todos los pasos indicados en Diagrama de flujo otorgado), Listar Usuarios (donde se generan listas ordenadas por usuario partiendo por su clave en el diccionario que es su RUT, seguido por los datos de la tupla que son los valores de cada clave, indicando nombre, apellidos y contraseña), Buscar Usuario (que sirve para encontrar usuarios dentro del diccionario ingresando su respectivo RUT), Eliminar Usuario (opción que elimina usuarios al ingresar su RUT para identificarlos), y Salir del programa.

Cabe destacar que se hicieron las distintas validaciones correspondientes:

* Al iniciar el Menú principal, si el usuario elije algún carácter que no es numérico, se captura la excepción con *Value Error* y se indica que debe ingresar un valor numérico. Si se ingresa un número fuera del rango de las 5 opciones también se informa que debe ingresar un número de 1 al 5, y vuelve a pedir el input.
* La opción Registrar Usuario / Crear Cuenta y sus validaciones ya es detallada en el Desarrollo del item anterior.
* La opción Listar Usuarios revisa si es que hay usuarios en el diccionario, y si no los hay, muestra un mensaje donde indica que el diccionario está vacío de usuarios y vuelve al Menú principal.
* La opción Buscar Usuario también valida si hay usuarios registrados antes de mostrar el input de búsqueda por RUT. Si el RUT ingresado no existe en el diccionario, muestra un mensaje de error donde indica que el usuario no existe, y vuelve al Menú principal.
* La opción Eliminar Usuario también valida la existencia de usuarios ingresados antes de mostrar el input de búsqueda por RUT. Si el RUT del usuario buscado para eliminar no coincide con los guardados en memoria, muestra un mensaje de error aclarando este hecho, y si existe el RUT en el diccionario, el usuario es eliminado junto con un mensaje de confirmación de la eliminación.

Este Menú principal es útil a la hora de revisar de forma ordenada los ingresos de usuarios, y permite realizar simples operaciones ‘CRUD’ a un nivel básico en memoria, que para el caso sirve para demostrar la correcta inserción de datos en una estructura de datos dinámica como lo es un diccionario, y operar sobre este durante el tiempo de ejecución del programa. Luego sería pertinente almacenar la información en una base de datos persistente como sería en una situación real de sistema.