

Automatización de procesos TIGO COLOMBIA B2B Gobierno

Automatización de procesos TIGO COLOMBIA B2B Gobierno.....	1
¿En qué consiste la automatización?.....	2
¿Cómo planear la automatización?.....	2
¿Aplica para automatización?.....	2
Definición del paso a seguir.....	2
¿Cómo pasar el plan de texto a código?.....	4
Explicación de Python.....	5
Descargar Python.....	5
Descargar Visual Studio Code.....	5
Números.....	6
Cadenas de texto.....	6
Variables.....	6
Índices.....	7
Listas.....	7
Diccionarios.....	7
If.....	8
For.....	8
Def.....	8
Explicación de librerías para Python.....	9
Librerías que vienen con Python.....	9
OS - SHUTIL.....	9
MATH.....	9
RANDOM.....	10
SMTPLIB.....	10
DATETIME.....	10
TIME.....	10
TKINTER.....	11
JSON.....	11
WEBBROWSER.....	11
PICKLE.....	11
SUBPROCESS.....	12
Librerías recomendadas para la automatización.....	12
PYPERCLIP.....	12
PYAUTOGUI.....	12
XLWINGS.....	12
PANDAS.....	12
MATPLOTLIB.....	13
PYNPUT.....	13
PYINSTALLER.....	13
Explicación del uso de ChatGPT para resolver problemas.....	13

¿En qué consiste la automatización?

La automatización consiste en utilizar tecnología para realizar tareas repetitivas para el ser humano. Está principalmente utilizada para mejorar la productividad de los empleados, mejora en el rendimiento de las tareas y reducción de costos.

¿Cómo planear la automatización?

¿Aplica para automatización?

Antes de todo hay que definir si la tarea debe ser automatizada, para saber esto la tarea debe tener alguna de estas características: límite de tiempo (Responder una solicitud lo antes posible), gran volumen (Gran cantidad de descarga de datos), repetitivo (Informes diarios, clasificación de datos) y acumulación (Más solicitudes de las que se pueden resolver). Si la tarea automatizar cumple con alguna o más de estas características, es óptimo empezar una automatización.

Definición del paso a seguir

Al momento de realizar la automatización de cualquier tarea se necesita realizar un plan para comprender cómo funciona realmente la tarea que se va a realizar. Es muy común que al momento de hacer una tarea se vea de una manera más “humana” y no se comprenda todo lo que hace parte de dicha tarea en términos de computación o robótica.

Lo primero que se debe hacer es explicar cada paso que se ejecuta al momento de realizar la tarea, la idea es ser lo más descriptivo y utilizar un lenguaje que sea sencillo de entender. Al principio no es necesario explicarlo completamente descriptivo, pero es necesario que se entienda lo suficiente, tanto como para seguir explicando cada paso o para realizar los “prompts” necesarios a ChatGPT para solucionar el problema.

Al comenzar solo se hace una explicación sencilla que después se va a estar expandiendo para que la tarea quede más dividida. Entre más dividida y explicada esté, la tarea será más fácil para ir parte por parte desarrollando la solución.

Por ejemplo:

“Descargar el archivo” \Rightarrow “Acceder a la página, hacer clic en el botón ‘descargar’, esperar la descarga”

“Extraer los datos” \Rightarrow “Acceder a la tabla X y traer los datos, guardarlos en el portapapeles”

Una técnica para identificar correctamente los pasos es hacer preguntas que expandan el paso, preguntas como: ¿De dónde? ¿Cómo? ¿De qué manera? ¿Cómo resulta el proceso?

Por ejemplo:

Descargar el archivo \Rightarrow ¿Cómo? \Rightarrow Haciendo clic en el botón “Descargar”.

Extraer los datos \Rightarrow ¿De dónde? \Rightarrow De la tabla seleccionada.

Actualizar la tabla dinámica \Rightarrow ¿De qué manera? \Rightarrow Seleccionando la tabla y actualizarla según la nueva tabla o los nuevos datos.

Extraer datos de una tabla \Rightarrow ¿Cómo resulta el proceso? \Rightarrow Tener dichos datos en el portapapeles o en otra tabla.

Es importante desarrollar la idea lo más que se pueda para que al momento de diseñar la automatización se tenga más claro cómo va a estar realizado el paso a paso.

¿Cómo pasar el plan de texto a código?

Teniendo ya terminado todo el plan en texto se puede empezar a pasar dicho texto al ámbito de programación. Como ya se conoce exactamente, QUÉ es lo que se necesita para que realice la automatización, ahora es necesario saber CÓMO lo va a realizar el programa de la mejor manera posible.

Así que, ¿Cómo se logra pasar el paso a paso de las tareas a un plan entendible en programación?

Primero se puede definir el lenguaje de programación que se hará uso, recomendando que sea Python, por varios motivos, como: gran acceso a librerías abiertas que facilitan las tareas, curva de aprendizaje sencilla y popular en procesos de automatización de tareas. Este documento está enfocado a explicar Python en la automatización de tareas, por lo que si no es tu caso, recomendando que busque más recursos para el lenguaje que desees utilizar.

Teniendo definido el lenguaje que se va a utilizar para la automatización de tareas, ahora se va a hacer lo posible para pasar cada paso que hicimos con anterioridad a funciones de Python. Más adelante se explicarán dichas librerías en mayor profundidad.

Primero se identificará en DÓNDE se está realizando dicha acción, esta va a permitir que se defina la librería que ayude a automatizar más fácilmente.

Por ejemplo:

“Descargar el archivo” ⇒ DÓNDE ⇒ En una página web.

“Extraer los datos” ⇒ DÓNDE ⇒ En una hoja de Excel.

Seguido a eso es importante identificar exactamente cómo se hace, con esto me refiero a cada movimiento del ratón y del teclado específicamente, esto para cada acción que se esté ejecutando en ese momento.

Por ejemplo:

“Descargar el archivo” ⇒ CÓMO ⇒ Acceder a la página web a través de un link, darle clic al botón de “Descargar”, esperar unos segundos.

“Extraer los datos” ⇒ CÓMO ⇒ Copiar los datos para cada persona que aparezca.

Con estos pasos definidos ya se puede empezar a buscar las librerías y realizar las funciones necesarias para la automatización.

Explicación de Python

Como se había comentado anteriormente, se recomienda el uso de Python para mayor comodidad, en este apartado se va a explicar lo básico que se debe conocer de Python para la automatización. En caso de querer abarcar más sobre este tema, se podrá hacer a través de la documentación oficial de Python, [Python Docs](#).

Descargar Python

Para poder utilizar Python se debe instalar desde la página principal, [Download Python](#), o desde la misma Microsoft Store. Esto descargará el intérprete de Python, y unas librerías básicas para empezar a programar.

Descargar Visual Studio Code

Es recomendable descargar Visual Studio Code para programar en Python, este editor de código permite extensiones que facilitarán la creación de aplicaciones de Python. Para descargarlo utiliza el link a continuación, [Download Visual Studio Code - Mac, Linux, Windows](#).

Números

Python reconoce los números e incluso puede ejecutar operaciones sencillas de los mismos en la misma consola.

```
2 + 3
5/2 # Las divisiones siempre muestran valores con un punto (.)
5//2 # Así aparecen las divisiones enteras
numero = 3 # Así se definen las variables
```

Puedes realizar operaciones y guardar números en variables, luego puedes utilizar estas mismas variables para aplicar más operaciones.

Cadenas de texto

Las cadenas de texto se pueden crear tanto con comillas dobles (""") como comillas simples ('). Puedes guardar dichas cadenas de texto en una variable. Para imprimir en la pantalla utilizas la función print() de Python.

```
print("Hola Mundo") # En la consola se muestra Hola Mundo
saludo = 'Hola Mundo' # Se guarda en una variable
print(saludo + ' :3') # Imprimirá el saludo junto con lo que se agregó
```

Variables

Cómo vimos antes se pueden hacer variables tanto de números como de cadenas de texto, estas variables pueden ser luego modificadas en función de lo que se necesite hacer.

```
saludo = 'Hola Mundo' # se crea la variable saludo
saludo = 5 # Se modifica dicha variable
print(saludo) # Va a mostrar el número 5
```

Índices

Los lenguajes de programación no comienzan a contar desde 1, empiezan desde 0, por lo tanto, el índice de listas, cadenas de texto, diccionarios y demás está definido por uno menos de lo que se tiene en cuenta.

```
saludo = 'Hola Mundo' # se crea la variable saludo
saludo[0] # equivale a la H del string
saludo[0:3] # imprime Hol, se excluye el valor 3
saludo[-1] # imprime la última letra del string (o)
```

Listas

Las listas se pueden crear como una variable, solo que el valor debe estar entre [], y una lista puede tener varios elementos. Es más, se puede guardar una lista dentro de otra lista, a este proceso se le conoce como anidación.

```
lista = ['Hola Mundo', 3] # se crea la lista
lista.append(':3') # se agrega al final de la lista
del lista[0] # elimina el valor 0 de la lista ('Hola Mundo')
```

Diccionarios

Los diccionarios en Python lo que hacen es guardar un valor por cada clave que tenga, igual que las listas, estos pueden ser anidados.

```
diccionario = {'Hola Mundo': 3, ':3': 'cara feliz'} # se crea el
diccionario
diccionario['adios'] = 'despedida' # así se agregan o modifican
valores al diccionario
```

If

If es una herramienta del control de flujo, sirve para tomar un camino u otro según las especificaciones que se han dado.

```
variable = 3 # se crea la variable
if variable == 3:
    print('La variable es 3') # salida si es 3
else:
    print('La variable no es') # salida si no es 3
```

For

For es un bucle, según las condiciones que se le coloquen va a realizar una función cada vez que se ejecute.

```
saludo = 'Hola Mundo' # se crea la variable
for letra in saludo:
    print(letra) # se imprime letra por letra el saludo
```

Def

Con def se crean funciones que luego se pueden utilizar en el aplicativo, funciones como las que hemos visto de print() y append().

```
def saludo(): # definiendo la función
    print('Hola Mundo') # contenido de la función

saludo() # ejecución de la función
```


Explicación de librerías para Python

Se explicará en este apartado las principales librerías de Python y las utilizadas en las automatizaciones, además, se compartirá la documentación de dicha librería para ampliar conocimientos. Las definiciones están enfocadas al aplicativo automaTIGO, así que es recomendable ampliar los conocimientos para tener mejor información.

Librerías que vienen con Python.

OS - SHUTIL

Estas dos librerías permiten la administración de archivos del computador, también para traer el path completo de los archivos que se encuentran en el computador y verificar si una carpeta o archivo existen. Para más información, aquí está la documentación oficial.

Os: [os — Interfaces misceláneas del sistema operativo — documentación de Python - 3.10.12](#)

Shutil: [shutil — Operaciones de archivos de alto nivel — documentación de Python - 3.10.12](#)

MATH

Como su nombre lo dice, esta librería trata de matemáticas, es una ampliación a lo que ya se puede hacer en Python con números, también permite funciones más complicadas de manera simplificada.

Math: [math — Funciones matemáticas — documentación de Python - 3.10.12](#)

RANDOM

Como su nombre lo dice, esta librería permite que se pueda seleccionar un valor aleatorio en una lista, string o diccionario.

Random: [random — Generar números pseudoaleatorios — documentación de Python - 3.10.12](#)

SMTPLIB

Esta librería permite enviar correos electrónicos desde Python.

Smtplib: [smtplib — Cliente de protocolo SMTP — documentación de Python - 3.10.12](#)

DATETIME

Esta librería permite que Python acceda al día y hora del computador, también permite traer semanas, meses y años anteriores exactos.

Datetime: [datetime — Tipos básicos de fecha y hora — documentación de Python - 3.10.12](#)

TIME

Esta librería permite que el programa espere un tiempo determinado en milisegundos hasta ejecutar la siguiente función.

Time: [time — Acceso a tiempo y conversiones — documentación de Python - 3.10.12](#)

TKINTER

Tkinter es una librería que permite generar interfaces de usuario que se ejecutan como ventanas y permite más accesibilidad a los usuarios.

Tkinter: [tkinter — Interface de Python para Tcl/Tk — documentación de Python - 3.10.12](#)

JSON

Esta librería permite codificar y decodificar archivos Json, estos son utilizados principalmente para el almacenamiento de variables del aplicativo.

Json: [Codificador y decodificador JSON — documentación de Python - 3.10.12](#)

WEBBROWSER

La librería de Webbrowser permite abrir links en el navegador.

Webbrowser: [webbrowser — Controlador de navegador web conveniente — documentación de Python - 3.10.12](#)

PICKLE

Esta librería permite manipular archivos pickle, estos archivos son de poco peso en donde se pueden guardar elementos de Python, tanto diccionarios, dataframes, listas, entre otros.

Pickle: [pickle — Serialización de objetos Python — documentación de Python - 3.10.12](#)

SUBPROCESS

Esta librería permite que se abra un archivo o carpeta.

Subprocess: [subprocess — Gestión de subprocesos — documentación de Python - 3.10.12](#)

Librerías recomendadas para la automatización.

PYPERCLIP

Esta librería permite que se copien datos al portapapeles.

Pyperclip: [pyperclip · PyPI](#)

PYAUTOGUI

Esta librería permite que Python interactúe tanto con hacer clics como oprimir teclas.

Pyautogui: [PyAutoGUI · PyPI](#)

XLWINGS

Esta librería permite la interacción con Excel desde Python, tanto ejecutar macros, actualizar y guardar libros como extraer datos de tablas.

Xlwings: [xlwings · PyPI](#)

PANDAS

Esta librería permite el mejor manejo de datos, ya sean listas, diccionario o dataframes.

Pandas: [pandas · PyPI](#)

MATPLOTLIB

Esta librería permite la visualización de gráficos a partir de datos previamente dados.

Matplotlib: [matplotlib · PyPI](#)

PYNPUT

Esta librería permite que se guarden las coordenadas de los clics dados por el usuario.

Pynput: [pynput · PyPI](#)

PYINSTALLER

Esta librería permite que se exporte un archivo .py para que funcione en cualquier computador (Windows) sin necesidad de importar todas las librerías desde 0.

Pyinstaller: [pyinstaller · PyPI](#)

Explicación del uso de ChatGPT para resolver problemas

Así como se debe definir las preguntas que permitan entender cómo se va a realizar la automatización, también es importante comprender cómo formular preguntas correctamente para que la herramienta de ChatGPT ayude a resolver cada paso de la automatización que se está realizando.

Se tiene que ser lo más específico con ChatGPT, esto para que desglose y comparta la respuesta que desde el principio se quería saber.

Por ejemplo:

Acceder a la página web a través de un link, darle clic al botón de “Descargar”, esperar unos segundos. ⇒

¿Cómo se accede a una página web a través de un link en Python?

¿Cómo se da clic con Python teniendo previamente unas coordenadas de la pantalla?

¿Cómo hacer que Python espere un momento antes de ejecutar una función?

Con esta definición de preguntas se puede encontrar mucho más fácil la respuesta. La idea también es probar desde varios enfoques e identificar qué es lo que se quiere lograr para a partir de dicha necesidad se forme otra pregunta a ChatGPT.