**Resumen Herramienta LaTex**  
<https://www.youtube.com/watch?v=bOIQ5Uphz6Y&t=267s&ab_channel=ComputerHoy.com>

Instalación y configuración de LaTex  
<https://www.youtube.com/watch?v=CovrDRDL81g&ab_channel=H%C3%A9ctorMisaelBacilioNavarro>

El código LaTex puede ser manejado por diferentes programas como Overleaf, Texmaker entre otros. La parte de creación del documento de salida de este proyecto se realizará con Texmaker. Una vez instalado, hay que buscar información de los comandos que se utilizaran, a continuación algunos omandos utilizados en este proyecto

1. \documentclass[12pt]{beamer}

Declara que el documento será de clase de diapositiva propio de LaTex, entre el corchete cuadrado se regula el tamaño de letra

1. \usepackage[utf8]{inputenc}

Codificación de entrada

1. \usepackage[spanish]{babel}

Packete de idioma español

1. \usepackage{amsmath}

Paquete de símbolos matemáticos

1. \title{Titulo del proyecto}

El títilo de la diapositiva “Titulo del proyecto”

1. \begin{document}

\end{document}

Estas dos etiquetas delimitan el contenido de nuestro documento. Es importante escribir siempre todo el contenido entre estas dos etiquetas. De lo contrario, será imposible compilar el documento.

1. \begin{Frame}  
   \end{Frame}

Dentro de un documento es habitual definir distintos entornos para crear bloques de contenido que LaTeX debe interpretar de forma distinta. Los entornos se definen siempre mediante las etiquetas, en este caso usamos frame para separar cada diapositiva e ingresar lo que llevará cada diapositiva en cada frame

1. \begin{itemize}

\item Texto

\end{itemsize}

Esto se usa para listar varios ítems en un párrafo

1. \newline

Agrega un salto de línea en la presentación

1. \hspace{}

nos permite definir el tamaño del espacio en blanco que deseamos insertar, dicho tamaño lo podemos definir en cm, por ejemplo, al escribir:

x\hspace{2cm}y

Tendremos:

x\hspace{2cm}y

Otra alternativa es escribir:

x\hphantom{12345}y

Tendremos:

x\hphantom{12345}y

Este último nos permite definir el tamaño del espacio en blanco según su argumento, por ejemplo, x\hphantom{123456789}y determina un espacio en blanco más grande que el anterior:

x\hphantom{123456789}y

**Llamado de Texmaker en Python**

Se requieren algunas consideraciones importantes antes de hacer la llamada desde Python:

1. Variables de entorno

Hay que configurar las variables de entorno dependiendo del sistema operativo. En este caso, para configurar la variable PATH en Windows:

Haz clic en el botón de Inicio y escribe "variables de entorno" en el cuadro de búsqueda. Selecciona "Editar las variables de entorno del sistema" en los resultados de búsqueda.

Se abrirá la ventana de Propiedades del sistema. Haz clic en el botón "Variables de entorno".

En la sección "Variables del sistema", busca la variable "PATH" y haz clic en "Editar".

Agrega la ruta de la carpeta donde se encuentra el ejecutable de Texmaker al final de la variable PATH. Asegúrate de separarla de las rutas anteriores con un punto y coma (;). Por ejemplo, si Texmaker está instalado en la carpeta "C:\Program Files (x86)\Texmaker", debes agregar esa ruta al final de la variable PATH de la siguiente manera:

; C:\Program Files (x86)\Texmaker

Haz clic en "Aceptar" en todas las ventanas abiertas para guardar los cambios.

Abre una nueva ventana de línea de comandos y escribe el comando texmaker. Si la variable PATH se configuró correctamente, el programa debería ejecutarse sin problemas. Reiniciar Python

1. Codificación utf-8

Para no tener problemas de codificación se puede escribir en la 1ra línea de Python el siguiente comando:

# -\*- coding: utf-8 -\*-

1. Versión de Python

Esta debe ser de la versión 3 en adelante

1. Hacer pruebas pequeñas probando ciertas partes del código

**Generar un archivo en Python**

Se utilizaron los siguientes comandos:

1. file= open(“beamer.tex”, “w”)

esta línea crea el archivo beamer de extensión tex, con w como write para empezar a escribir en el archivo. File es la variable que manejará Python para comunicarse y exportar el documento

la letra w se puede reemplazar por una a en caso de que se quiera agregar líneas al archivo generado en una instancia anterior

1. file.write(" Z ")

comando para ingresar un string o una variable(sin comillas). Es importante recalcar que este comando solo acepta una única entrada, por lo que cada vez que se quiera meter mas de un string o variable se debe volver a escribir el comando. El string “\n” hace un salto de línea en el documento generado

1. file.close()

cerrar el archivo abierto previamente

**Importante:**

Debido a que Latex utiliza el backslash “\”” al empezar todos sus comandos de línea, en Python

codigo en Python se añade una r antes del string. En Python, las "raw strings" y las cadenas normales son esencialmente lo mismo, es decir, son objetos del tipo **str**. La única diferencia es que las "raw strings" se representan con el prefijo **r**, lo que hace que los caracteres de escape (como **\n** o **\t**) se interpreten literalmente en lugar de su significado normal. Otra opción es añadir “u”, pero este método no resultó.

**Para llamar un programa externo se necesitará un comando de la biblioteca subprocess**

Primero se debe llamar a la biblioteca

Import subprocess

El comando se llama .call y se utilizó de la siguiente manera:

Subprocess.call([“texmaker”, “beamer.tex”])

Donde texmaker es el nombre del programa que utiliza latex y beamer.tex es el archivo generado por Python en las líneas anteriores

**Importante:** el comando .call también puede enviar líneas de comando en vez de un archivo entero. Aprovechando esa funcionalidad se quizo implementar el código y enviarlo directamente a Texmaker en vez del archivo .tex, pero al parecer genera conflictos con el raw string incluido en el código anterior, por lo que esta opción se desechó.

En caso de que se desee converir un raw string a un string normal basta con declararlo en una nueva variable de la siguiente manera, donde rawstring es lo que deseamos convertir

Variables = str(rawstring)