**En el codigo paso01\_extraer\_texto\_df\_ppts.py** se usa la presentación originales de forma que para cada slide de cada archivo se guarda un archivo en PPTs\_elementos/{archivo}/slideNNN\_content.txt que denomino la “descripción técnica de la slide”, con un formato así:

A continuación se listan las componentes de texto e imagen de la slide

Hay un grupo de 2 elementos de texto de relleno y borde sin color cuya posicion relativa es "cobertura completa", y los textos de arriba a abajo son:

"Consideraciones"

"La suspensión por Extravío genera una clave de recuperación que permite al usuario solicitar posteriormente la reactivación. Si el usuario no tiene la clave de recuperación para reactivar la línea, la reactivación sólo puede ser solicitada por un contacto con relación crítica.

Para reactivaciones por extravío de líneas migradas en dicho estado, el sistema NO solicitará la clave de recuperación.

Para la reactivación de línea se debe tener presente ingresar un SR con la categoría y motivo correctos, que deben coincidir con el motivo de la suspensión, caso contrario, aparecerá un error en la orden.

(\*) Los datos mostrados en las pantallas corresponden a los disponibles en ambiente de prueba (UAT), por lo que son referenciales y podrían tener alguna modificación menor en el futuro

"

----------------------------------------

Hay un texto "Entel 2023" de relleno y borde sin color cuya posicion relativa es "abajo-centro"

----------------------------------------

Hay un texto "3" de relleno y borde sin color cuya posicion relativa es "abajo-derecha"

----------------------------------------

Tambien están en la slide las siguientes imagenes:

Imagen imagen1.wmf cuya posición relativa es "arriba-izquierda"

Imagen imagen2.wmf cuya posición relativa es "completo derecha"

Imagen imagen3.png cuya posición relativa es "arriba-derecha"

------------

Por ahora el formato está relativamente bien así que no lo voy a cambiar aunque sería bueno por ejemplo describir las imágenes antes de los textos ya que en estos se mencionan las imágenes, lo cual es un PROBLEMA.

En este código también se guardan las imágenes (incluyendo repetidas) de la presentación original con los paths/nombre PPTs\_elementos/{archivo}/slideNNN\_imageNN.png que denomimanos la “imagen de la slide”.

En este código también se guarda una imagen por slide de cada archivo con el path/nombre PPTs\_separados/{archivo}/slideN.png lo que denominamos la “slide original”. El PROBLEMA es que el número de dígitos que se usan para al número de la slide sube para presentaciones de más de 100 paginas lo cual es inconveniente para los códigos que se usan más adelante.

Si lo corro de nuevo mejor anotar cuanto tiempo demora porque es un PROBLEMA no saberlo.

**Luego el código paso02\_borra\_imagenes\_repetidas.py** borra las imágenes repetidas en las carpetas PPTs\_elementos/{archivo} para ahorrar espacio usado innecesariamente y crea un json imágenes\_borradas indicando las que se borraron y cuál es su contraparte aún existente.

**Luego el código paso03\_crea\_csvs\_contenido.py** crea los archivos que irán a la base de datos para ser usados para elegir contexto y responder las preguntas de los usuarios.

Por cada archivo, primero se crea la carpeta Informacion\_Archivos/Informacion\_Paginas/{archivo}/

Y se usa como input las descripciones técnicas de todas las slides, el nombre del archivo,

en la función hace\_prompt\_resumen\_preliminar\_archivo

y Chat GPT para crear el “resumen preliminar del archivo”.

Luego para cada slide se usa el resumen preliminar del archivo, los resúmenes de las slides anteriores a la actual, la descripción técnica de la slide, el nombre del archivo, y el número actual y total de slides. Con esto de input en la función hace\_prompt\_descripcion\_slide y la imagen de la slide en la función hacer\_pregunta\_con\_imagen con Chat GPT creamos el archivo Informacion\_Archivos/Informacion\_Paginas/{archivo}/Informacion\_SlideNNN.txt que denominaremos la “descripción completa de la slide”.

Luego se usa la descripción completa de la slide, la descripción técnica de la slide, el nombre del archivo, y el número actual y total de slides. Con esto de input en la función hace\_prompt\_resumen\_slide y Chat GPT obtenemos el “resumen de la slide”. Este resumen se usa tanto para hacer el resumen de las slides anteriores para crear las descripciones completas de las slides, y también se guarda en el “json de todas las páginas” que se guarda como un solo archivo para toda la base de información con el nombre Informacion\_Archivos/paginas.json.

Luego los resúmenes preliminares de todos los archivos se guardan juntos en Informacion\_Archivos/archivos.json en lo que denominamos el “json de todos los archivos”.

Este código demora alrededor de media hora para archivos de ~80 slides, donde el tiempo sube un poco más rápido que lineal, y gasta alrededor de 400 K tokens (~6 centavos de dolar) para un archivo así.

**Luego el código paso04\_creates\_database.py**, crea la base de datos donde iran las tablas asociadas a la base de información

**Luego el código paso05\_creates\_tables.py** crea las tablas archivos, paginas, respuestas y fallos para guardar la información que será usada para determinar la estrategia a usar para responder la pregunta (pregunta nueva, tema encontrado, pregunta encontrada), para determinar el contexto necesario (archivo y rango páginas) para responder las preguntas, y para usar de prompt para responder la pregunta del usuario. Además respuestas y fallos guardan información sobre las preguntas hechas de forma exitosa y fallida respectivamente.

**Luego el código chat\_app\_v0N.py** (la versión va cambiando con el tiempo) se encarga de correr la app para responder las preguntas.

Primero para cada pregunta creamos un objeto de la clase Pregunta a partir de la pregunta del usuario y el contenido actual de la tabla respuestas y fallos. Los atributos de este objeto guardan la información de la pregunta hecha, y la información asociada a cada una de las etapas del proceso, determinar estrategia (pregunta nueva, tema encontrado o pregunta encontrada), determinar contexto (archivo y páginas) y respuesta final.

Una vez creado el objeto usamos el método determina\_estrategia para determinar si vamos a usar la respuesta de una pregunta existente para responder la pregunta (pregunta encontrada), si vamos a usar el contexto de una pregunta existente para responder la pregunta (tema encontrado) o si vamos a buscar el contexto y la respuesta en la base de información (pregunta nueva).

Luego según la estrategia determinada, usamos el método determina\_contexto para determinar el archivo y las páginas a utilizar para responder la pregunta. Si la estrategia es pregunta encontrada o tema encontrado simplemente tomamos el archivo y el rango de páginas a usar de los usados por la pregunta similar encontrada en las existentes. Si la estrategia es pregunta nueva, usamos los resúmenes preliminares de cada archivo en el dataframe de la tabla archivos, más la pregunta del usuario en el método hace\_prompt\_elige\_archivo, Chat GPT y luego el método determina\_nombre\_archivo\_a\_usar para determinar el archivo a usar como contexto. Si este proceso es fallido se aborta la respuesta y se declara fallida, si es exitoso usamos el nombre del archivo usado, los resúmenes de las slides del archivo a usar del dataframe de la tabla paginas y la pregunta del usuario como input del método hace\_prompt\_elige\_paginas. Luego con Chat GPT y el método determina\_paginas\_elegidas determinamos las páginas a usar como contexto.

Luego según la estrategia y el contexto (archivo y páginas) usamos el método determina\_respuesta para determinar la respuesta final al usuario. Si la estrategia es pregunta encontrada usamos la respuesta de la pregunta similar existente. Si la estrategia es pregunta nueva pero no pudimos determinar el contexto se aborta la respuesta y se declara fallida. Y si la estrategia es tema encontrado o fue pregunta nueva y se pudo determinar el contexto, entonces hacemos la pregunta del usuario a Chat GPT usando el contexto determinado anteriormente.

Para esto último usamos la descripción completa de las slides involucradas, el nombre del archivo, y la pregunta del usuario como input del método hace\_prompt\_final. Luego con Chat GPT determinamos la respuesta final al usuario.