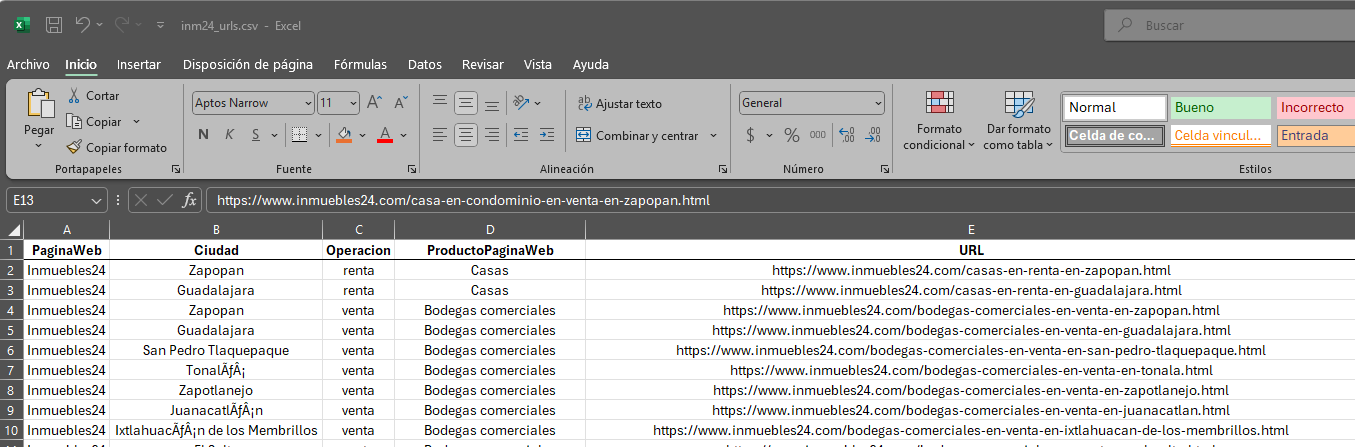
Analiza a profundidad todo el proyecto, lee "Objetivo Principal.txt" para entender mi objetivo principal, luego lee "Objetivo Orquestacion.txt" para saber el objetivo del proyecto este es el que hay que modificar para el funcionamiento del sistema al igual que todos los scripts. También lee “Estudio.txt” y Revisa conexiones, referencias, inputs y outputs de todos los scripts de orquestación, utilidades, configuraciones, Scrapers. Implementa todos lo arreglos necesarios y corrige documentación.

Flujo Scrapers:

Cada Scraper en la carpeta “Scrapers” tiene su respectivo archivo csv en la carpeta de “urls”, necesito que todos los scrapers y la orquestación con todas sus variables se guíen con estos csv. Cada Scraper entra un url y dentro de ese url recorre todas las páginas existentes, hay url con 1 página o con 200 páginas, o ninguna página.



<Variables\_Orquestacion>

The composition of the CSVs in the "URLs" folder defines the hierarchy that the orchestration must follow.

<Column1 name="PaginaWeb">Website: (Inm24, CyT, Lam, Mit, Prop, Tro)</Column1>

<Column2 name="Ciudad">City: (Gdl, Zap, Tlaj, Tlaq, Ton, Salt, Zptl, IMem, Jnctl)</Column2>

<Column3 name="Operacion">Operation: (Ven, Ren, Ven-d, Ven-r)</Column3>

<Column4 name="ProductoPaginaWeb">ProductWebsite: (Dep, Cas, Ofc, Com, etc…)</Column4>

<Column5 name="URL">Extraction URL (this column marks the order that the scrapes must follow according to the order of the URLs in the CSV; it is important to maintain this order) </Column5>

</Variables\_Orquestacion>

Necesito que en toda la orquestación, scrapers y todos los scripts usen abreviaturas, coloque un csv “Lista de Variables/Lista de Variables Orquestacion.csv” con las abreviaturas, columna 1 la variable y Columna 2 como se debería de abreviar.

Ejemplo de flujo:

El script “Inm24.py” busca en carpeta Urls, abre “inm24\_urls.csv” lee la celda Fila 2 y Columna 5, toma url: “https://www.inmuebles24.com/casas-en-renta-en-zapopan.html”, luego entra a:  
“https://www.casasyterrenos.com/buscar/jalisco/zapopan/departamentos/venta?desde=0&hasta=1000000000&utm\_source=results\_page%3D%22&page=1”, toma información y luego va a “https://www.casasyterrenos.com/buscar/jalisco/zapopan/departamentos/venta?desde=0&hasta=1000000000&utm\_source=results\_page%3D%22&page=2”, toma información y así hasta “https://www.casasyterrenos.com/buscar/jalisco/zapopan/departamentos/venta?desde=0&hasta=1000000000&utm\_source=results\_page%3D%22&page=N”, la estructura de como pasar del url original a la estructura para el recorrido de páginas o bucle viene descrita en “cyt.txt” (cada Scraper tiene una forma de recorrido o bucle diferente y tiene su propio documento txt) en la carpeta “Scrapers Originales”:  
“””  
for i in range(n, total\_pages + 1):

URL = f"{URL\_BASE}{i}"

driver.get(URL)

... scrape\_page\_source(driver.page\_source)  
“””  
Después el Scraper guarda toda la información en un csv y en sql lite,para definir el nombre del csv y la carpeta donde se guardara el orquestador necesita ver todas las variables en el csv de Url, una vez que el Scraper acabe con todas las páginas, vuelve a leer su archivo csv de Url y va por el siguiente Url (Fila 3 y Columna 5).

En el caso de inm24 es diferente:

El script “Inm24.py” busca en carpeta Urls, abre “inm24\_urls.csv” lee la celda Fila 2 y Columna 5, toma url: “https://www.inmuebles24.com/casas-en-renta-en-zapopan.html”, luego entra a:  
<https://www.inmuebles24.com/departamentos-en-venta-en-zapopan-pagina-1.html>, toma información y luego va a <https://www.inmuebles24.com/departamentos-en-venta-en-zapopan-pagina-2.html>, toma información y así hasta <https://www.inmuebles24.com/departamentos-en-venta-en-zapopan-pagina-N.html>, este Scraper solo toma Urls de publicaciones en las paginas que recorrió, hace un csv puente con el que alimenta a “Inm24\_det.py” y es este es quien generara la información de las propiedades, pero el flujo debe de contemplar siempre que acabe inm24 debe de iniciar inm24\_det. Y pasa lo mismo con lam y lam\_det.

El scraper principal son los de Inmuebles24 este siempre tiene que estar corriendo hasta que se acabe sus urls, paralelo a inm24 tiene que haber corriendo al mismo tiempo 3 scrapers de las otras Paginas Web y tienen que ir intercambiando de una pagina web a otra cada vez que acaben un url.

Estos ejemplos son solo para retratar el flujo de trabajo, el flujo puede ser llevado por el mismo scraper o la orquestación, lo que sea mas optimo y funcional. Es muy importante que contemples que los scrapers son procesos de horas y necesito un buen sistema que pueda iniciar donde se quedó el trabajo después de algún apagón o desconexión de internet, seguimiento de trabajo tanto de la orquestación como de los scrapers.