**“COMPARADOR DE TARIFAS”: DOCUMENTACIÓN**

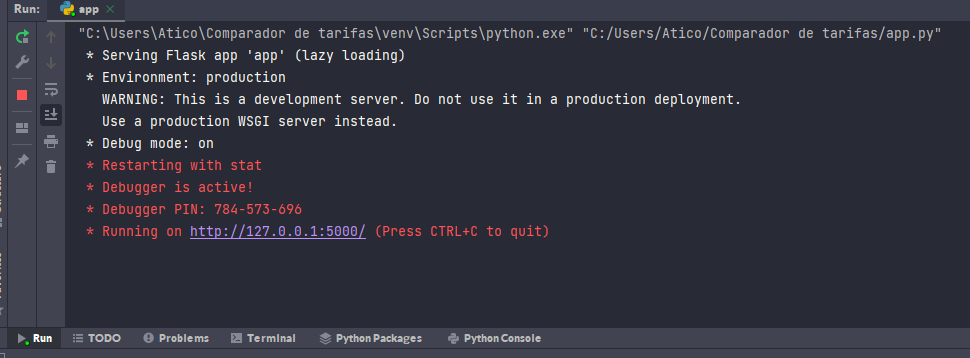
**Presentación:**  Se trata de una web comparadora, en este caso de tarifas de luz de mercado libre ibérico, al estilo de otras como Rastreator.com, en las que además te dice cual es la más económica dada una situación particular (tu consumo eléctrico)

Esta aplicación ofrece información de 5 tarifas distintas de diferentes compañías (Naturgy, Iberdrola, Endesa, Repsol y EDP), las 5 más importantes en España. Aunque hay ligeras diferencias por zona geográfica, compañía distribuidora y servicios que pueda incluir la factura, esta se resume en dos términos: potencia y consumo.

La aplicación automatiza la búsqueda del precio del término de consumo, el cálculo de la factura dados los datos de consumo mensual y potencia contratada y ofrece estadísticas sobre el precio de factura sobre la del resto de clientes de la compañía (clientes ficticios generados automáticamente por Python, no hay acceso a los datos de clientes reales)

**Funcionamiento:**

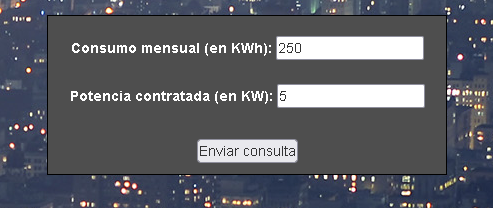
Ejecutamos app.py en PyCharm



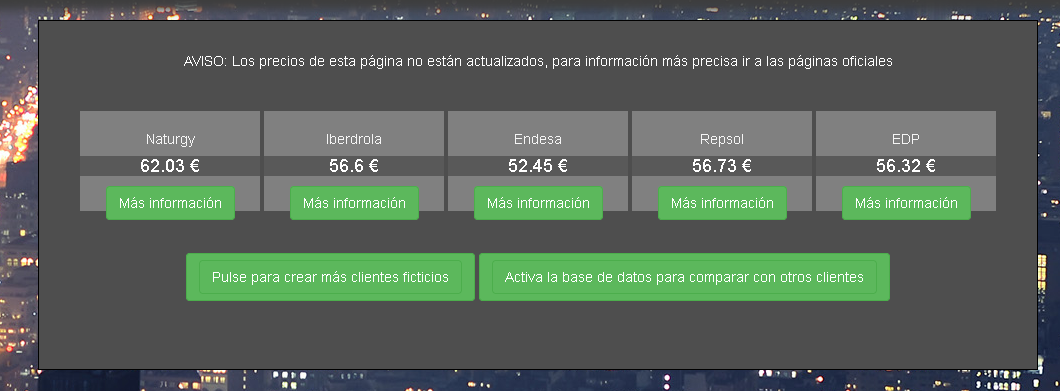
Clickamos en el link que aparece en terminal y se nos abre esta página en el explorador



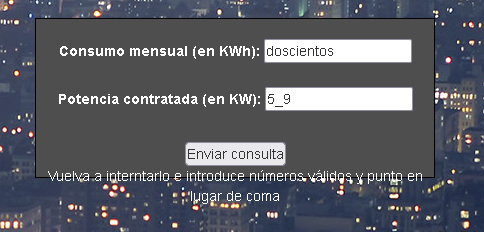
Elegimos unos datos razonables para un consumo (mensual) y una potencia eléctrica



Al enviar consulta, cambiamos a una página en la que nos aparece dicho la factura debida a ese consumo en varias tarifas de distintas compañías



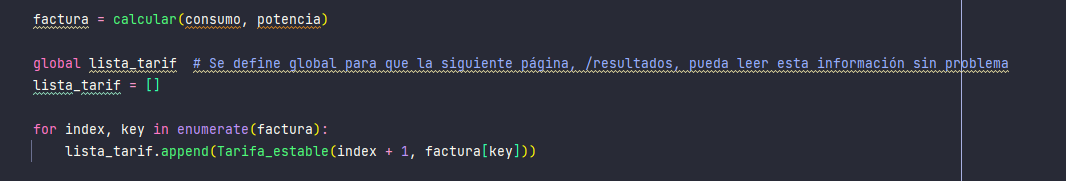
Si introducimos algo que no sean números o escribimos comas en vez de puntos, un mensaje de error nos devuelve a la página original



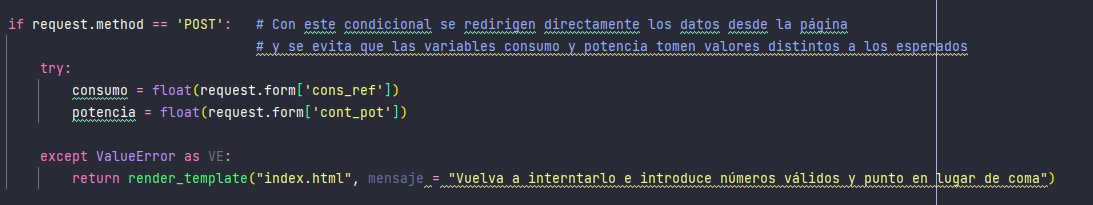
Lo que hacemos en dentro del programa es rellenar un formulario que lleva a la ruta calculo a través de un método POST



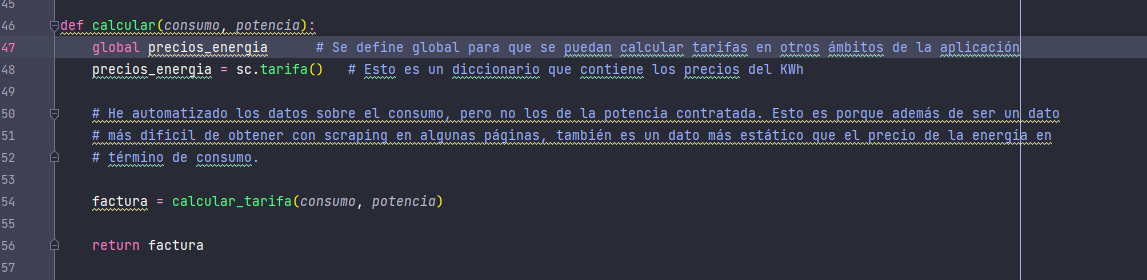
Los parámetros introducidos consumo y potencia se meten dentro de una función



Recepción de los datos del formulario y control de excepciones por si se introduce un valor no válido



precios\_energia devuelve los precios del KWh según algunas tarifas que he escogido. sc. es un import de scraping.py, el cual explicaré luego



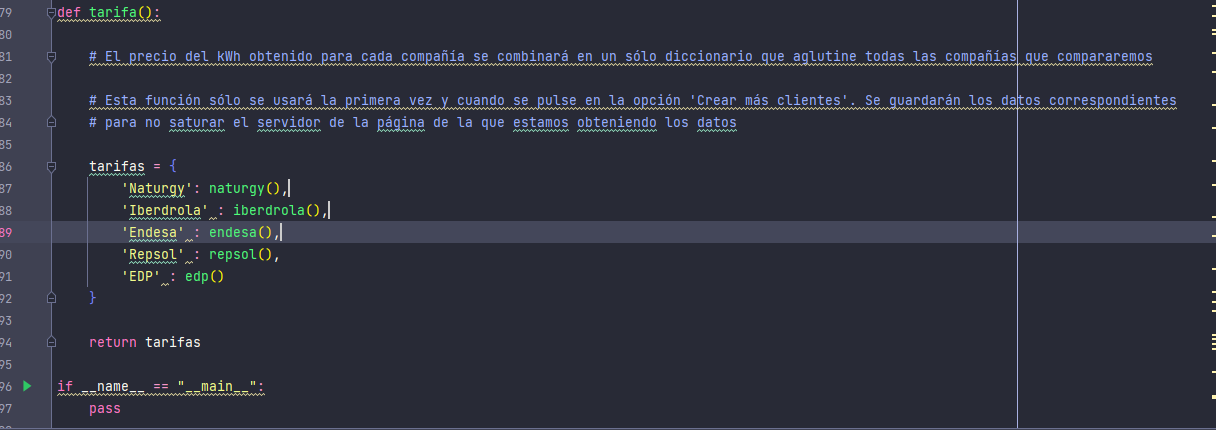
calcular\_tarifa es el núcleo del programa, donde se juntan todos los datos anteriormente obtenidos (precio KWh, precio KW potencia, consumo y potencia) para construir un diccionario de valores numéricos que correspondería al precio de la factura

En scraping.py, definimos una función para cada tarifa que queremos analizar. El funcionamiento es similar para el resto de tarifas, cambiando el xpath según la página

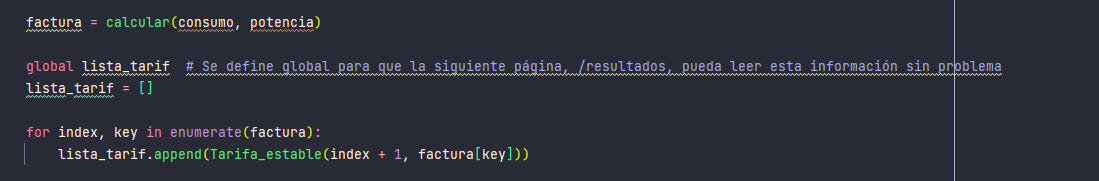
Texto

Descripción generada automáticamente

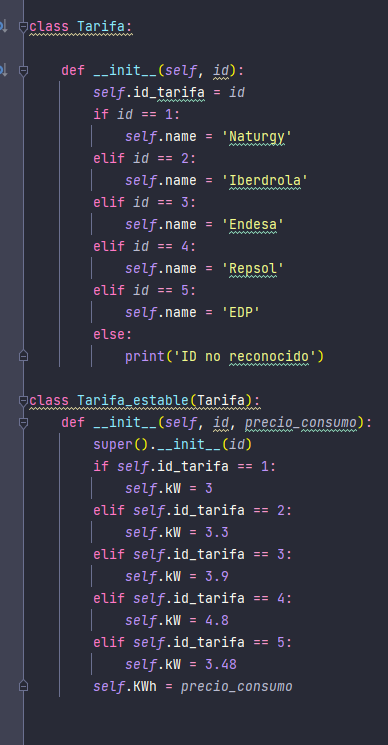
La función tarifa de scraping.py genera un diccionario de precios, que es el return de todas las funciones anteriores



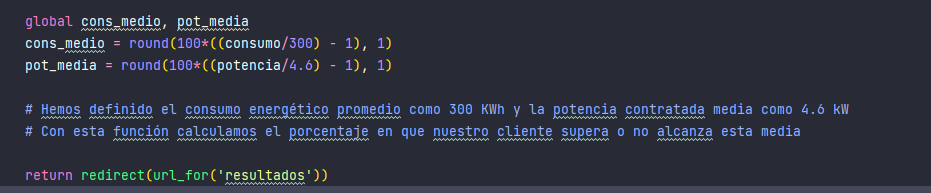
El primer factura es local, así que factura global, el diccionario anteriormente definido, tiene prioridad en el bucle. Se va generando una lista de objetos tarifa\_estable



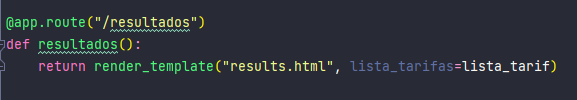
La clase tarifa\_estable hereda de tarifa un id y un name, para identificar cada compañía de manera más sencilla. Una vez establecido el id, se le asocia un valor al precio de la potencia y a la del consumo

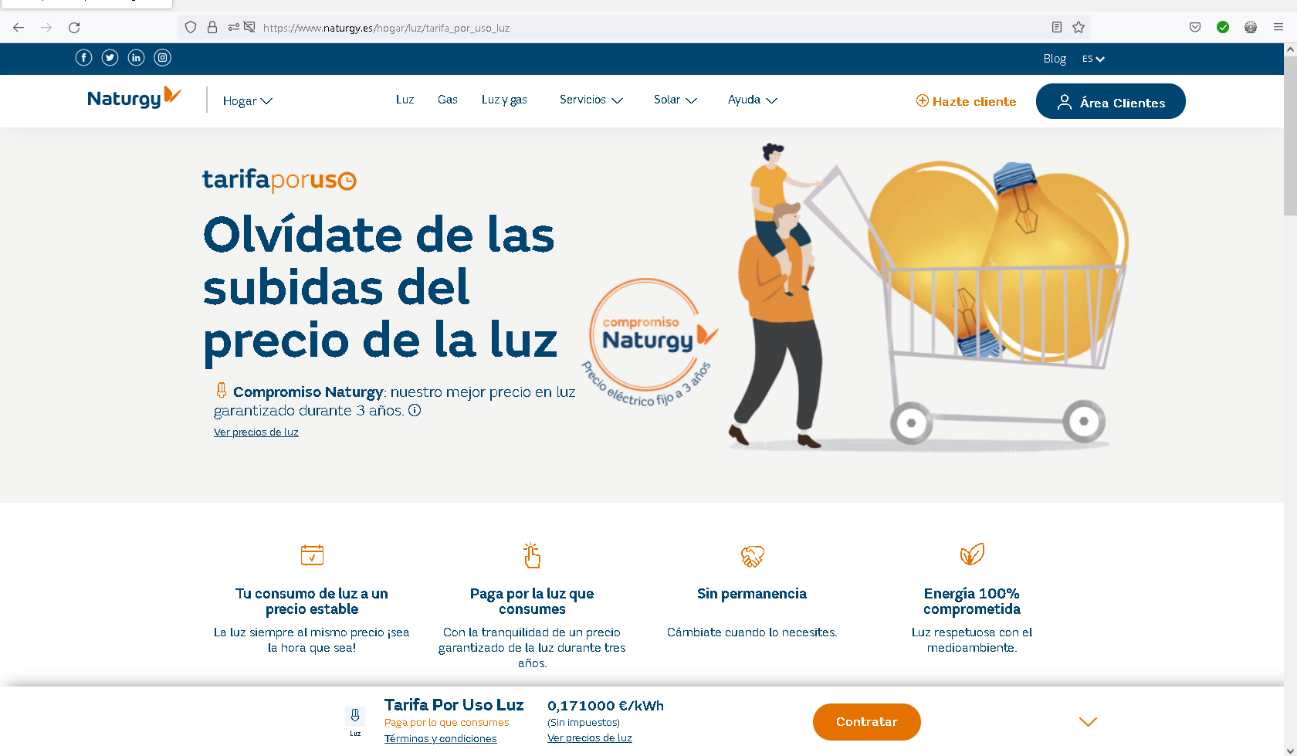


Con estas variables más tarde haremos estadísticas sobre el consumo y la potencia



Se redirige a resultados, que tan solo genera la segunda página que ya hemos visto, pasandole como información la lista de objetos tarifa que hemos rellenado en el bucle



Al hacer click en cada botón de Más información, nos redirige a la página oficial en la que aparece la tarifa que se ha estado calculando

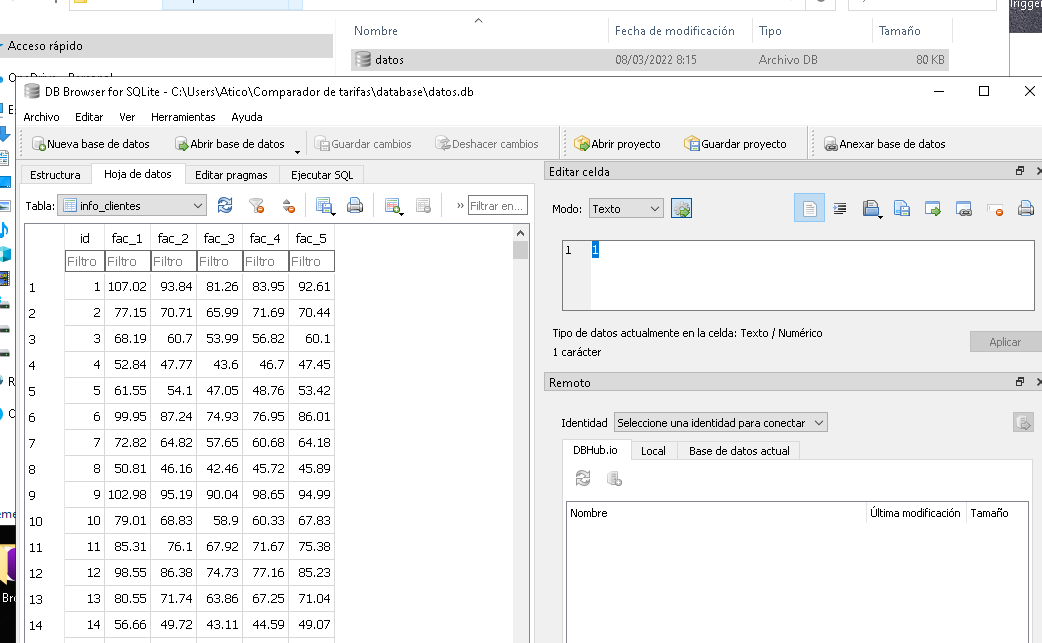
Más ejemplos de páginas oficiales. Salvo Naturgy el resto de los precios no están actualizados porque estas páginas usan JS. En próximas versiones se implementará la ejecución de JS



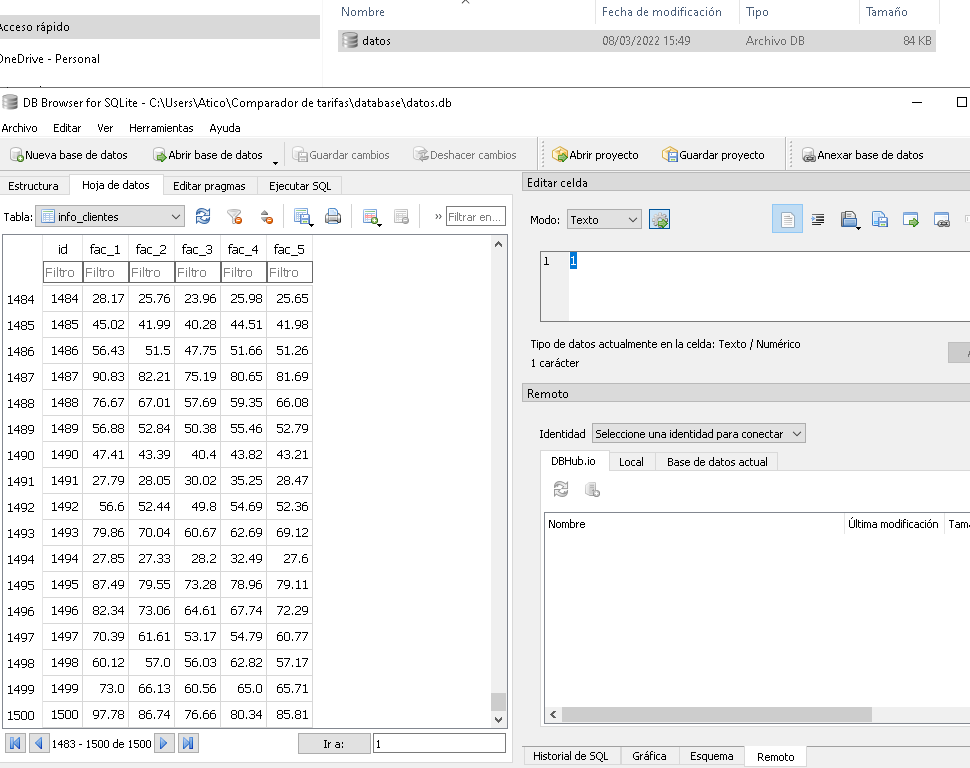
A nivel código, esta lista de precios la he hecho con un bucle Jinja para cada una de los elementos en lista\_tarifas. Además, dentro del hrc hay un condicional Jinja que decide a qué página enlaza cada botón



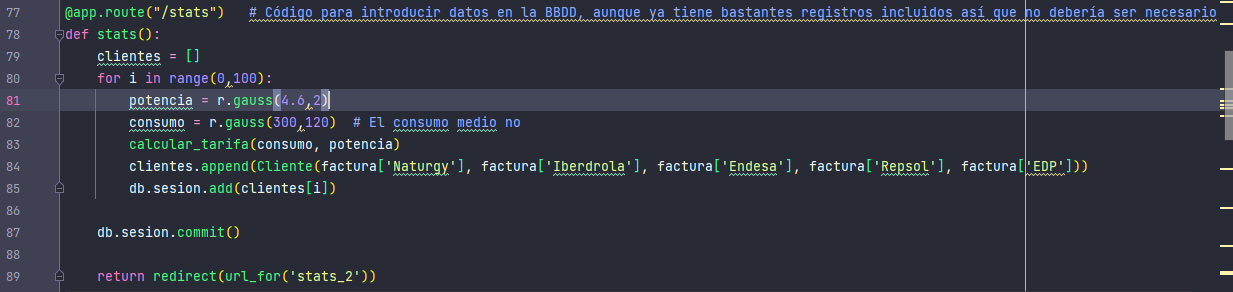
Aquí aparece la base de datos por primera vez, a la cual ya le he añadido bastantes registros de los cuales hablaremos a continuación



Al pulsar el botón de añadir más clientes ficticios, se añaden 100 registros más a la base de datos y crece en 4 KB



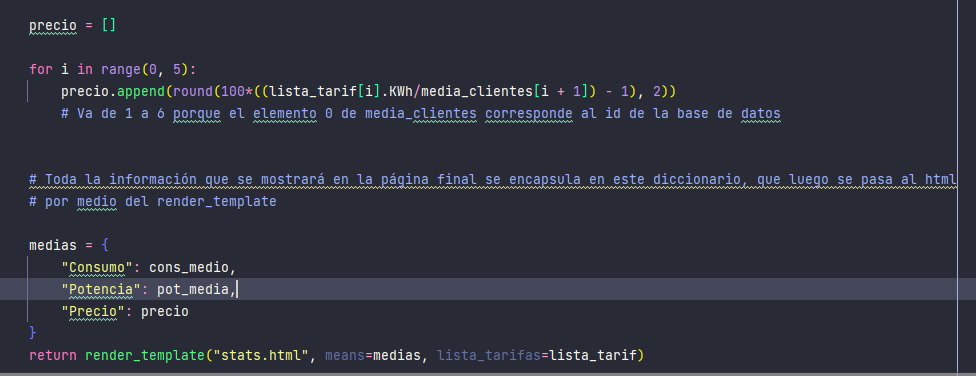
Se crea una lista de clientes vacía, se generan números aleatorios basados en una distribución gaussiana con cierta media y desviación típica, se les aplica la funcion calcular\_tarifa y se generan objetos cliente



Cuando termina el bucle se hace un commit, se guarda en la base de datos y se cambia a la página final. Si en la página results se pulsa en 'Activar base de datos' el resultado es el mismo sin añadir 100 registros más



Después de conectarme a la base de datos, extraigo las medias de cada columna, que representan el precio medio de clientes de esa compañía. Luego calculo el porcentaje entre el precio que nos sale a nosotros con esta media



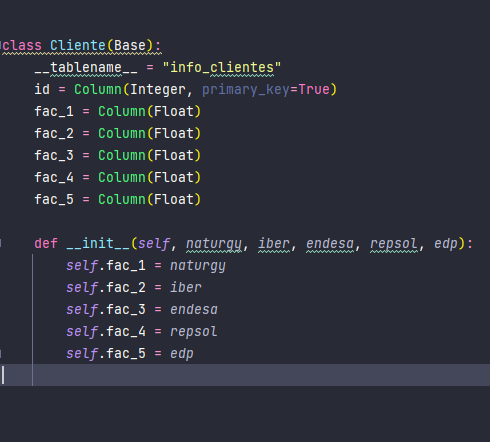
La fase final de la aplicación se vería así, usando Jinja también en esta parte para acceder a variables definidas en Python



Si se pulsa el botón de la página principal se vuelve al comienzo

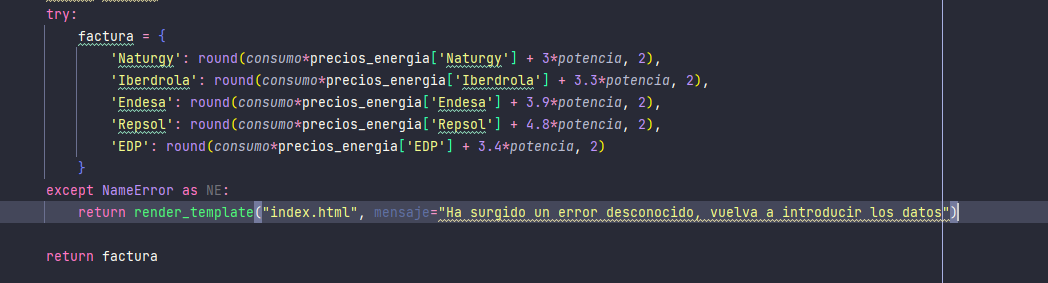


Clase Cliente que se guarda en la base de datos. Es tan sólo un registro de precios por compañía para un mismo cliente simulado



**Mejoras para próximas versiones:**

* En ocasiones se encuentra un error al pulsar en el botón de añadir más clientes ficticios después de haber pulsado en los de Más información. De momento lo soluciono con este try except que devuelve a home



* Salvo la de Naturgy, el resto de tarifas están sacadas de páginas desactualizadas por un sencillo motivo: la información en las páginas oficiales se encuentra dentro de desplegables y elementos interactivos JavaScript, lenguaje que no sé programar. Se implementará en próximas versiones
* Para no alargar la aplicación, se han tomado en cuenta sólo 5 tarifas de precio estable y de mercado libre. En un futuro, la aplicación puede abarcar más tarifas dentro de estas compañías, más compañías (en España hay más de 200 comercializadoras eléctricas, aunque estas sean las 5 mayoritarias), tarifas planas, otros tipos de tarifas, tarifas de mercado regulado e incluso, para versiones muy posteriores, analizar las tarifas que se ofrecen a empresas
* Casi todas las compañías que ofrecen luz también comercializan gas, y hay tarifas en las que contratar ambas con la misma compañía genera ciertos descuentos. Implementar el cálculo de las tarifas de gas es otra opción valorable en el futuro