# Introducción y objetivo

CNMWebOrdersScrapper.py es un módulo programado en Python utilizando principalmente la herramienta *Selenium* y creado específicamente para la automatización del acceso a la página de compras de IMB-CNM-CSIC con el fin de recopilar los datos correspondientes a las órdenes de compra de un determinado usuario para un determinado periodo de tiempo determinado por un año inicial y un año final.

# Motivación

La motivación principal para el desarrollo de esta herramienta es resolver, asumiendo que no podemos hacer uso de consultas sql a la base de datos, el problema que representa la consulta en el histórico de órdenes de compra vía intranet del IMB-CNM. Vía intranet del IMB la consulta del histórico de compras de un usuario es un proceso tedioso y falto de herramientas de filtro y/o búsqueda de determinadas órdenes de compra (p.e filtro por proveedor, búsqueda de un determinado material las líneas de las órdenes de compra, etc..).

# Como se usa

*CNMWebOrdersScrapper* es una herramienta que define una serie de métodos que solo funcionan bajo un determinado contexto que viene marcado principalmente por la página web en la que se encuentra el navegador en el momento de la llamada a dichos métodos. Por ejemplo, no sirve de nada llamar al método *getOrderData* si no nos encontramos en la página web correspondiente al detalle de la orden de compra donde aparece la lista de artículos (líneas) de dicha orden de compra.

Podríamos decir que se trata de un tipo de programación contextual.

# Donde se usa

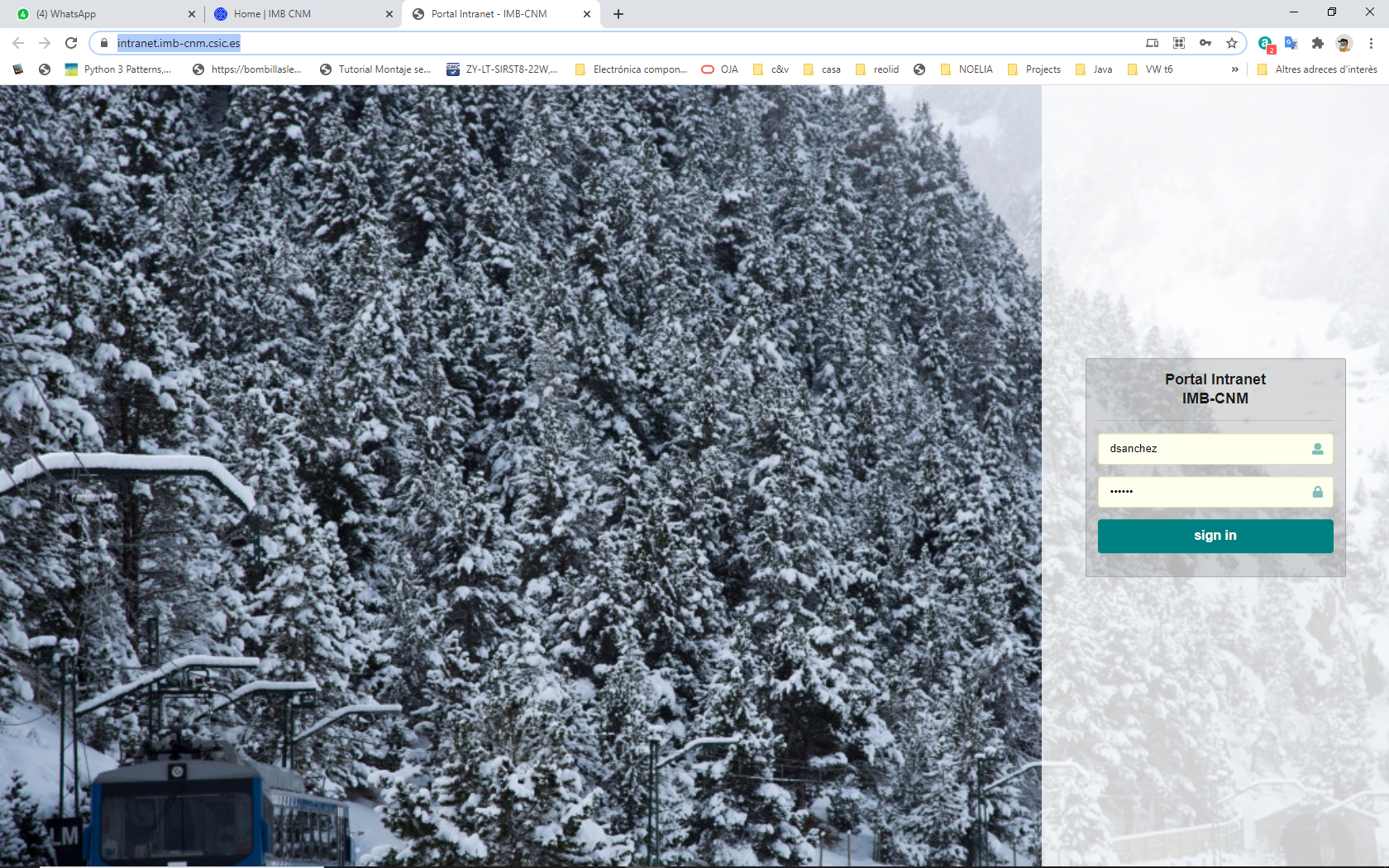
Para el uso de la herramienta *CNMWebOrdersScrapper* se ha creado un case principal llamada *getCNMOrders* que es la ejecutada para recuperar los datos de las órdenes de compra, dicha clase debe recibir como parámetros de entrada lo siguiente.

Parámetros de entrada de *getCNMOrders*

* *Posición 1*. Login\_url

Url de la página para el acceso a la intranet del IMB-CNM.

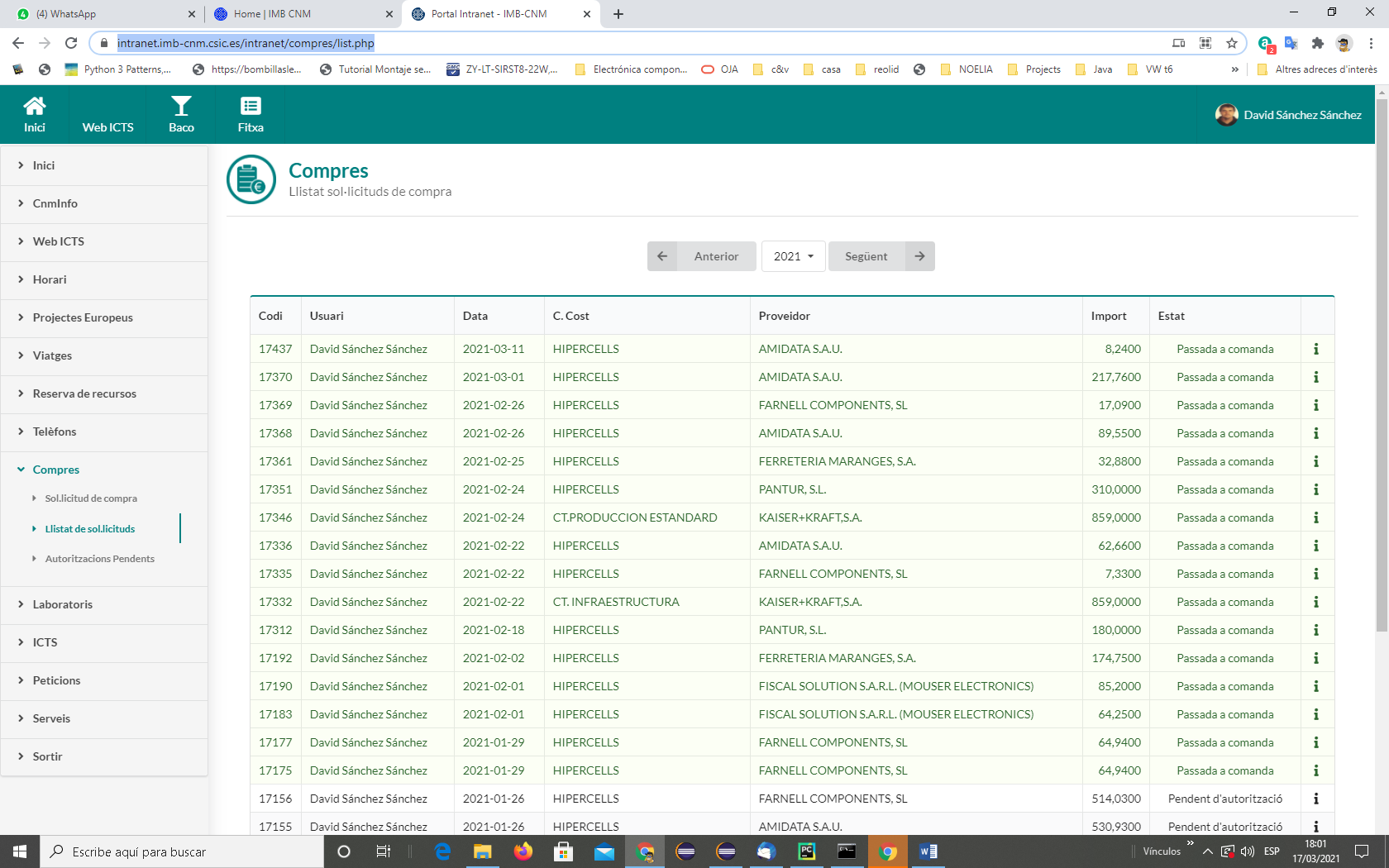
Actualmente es <https://intranet.imb-cnm.csic.es/>



* *Posición 2*. orders\_url

Url de la página con el listado de ordenes de compra del ultimo año en curso.

Actualmente es <https://intranet.imb-cnm.csic.es/intranet/compres/list.php>



* *Posición 3*. User

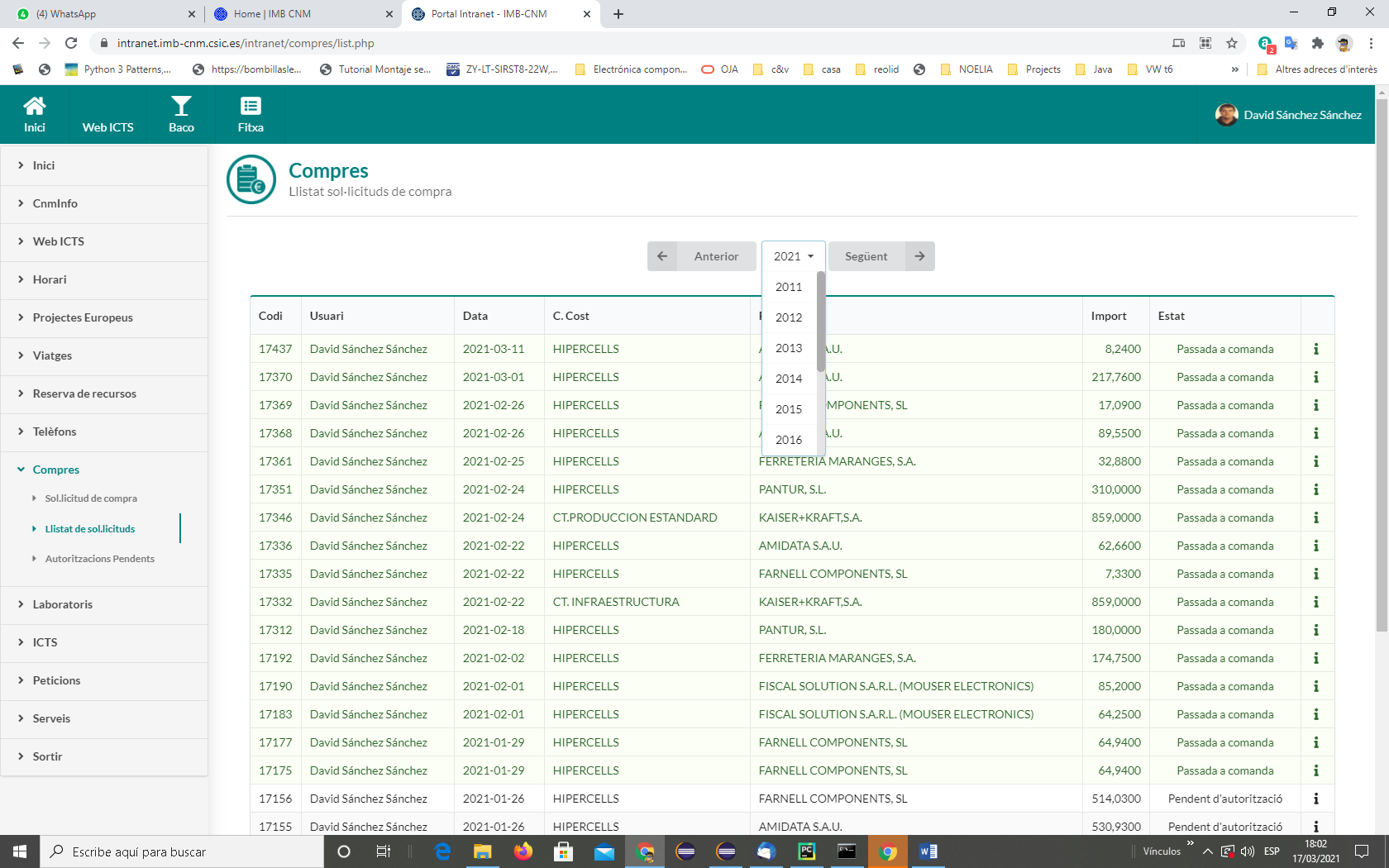
Nombre de usuario para el acceso a la intranet del IMB.

* *Posición 4*. Pass

Contraseña de usuario para el accesi a la intraner del IMB.

* *Posición 5*. First\_count\_year

Primer año que ofrece registros de compra la página web del IMB.



* *Posición 6*. From\_year

Primer año, incluido, desde el cual getCNMOrders comenzará abuscar las ordenes de compra del usuario ern cuestión.

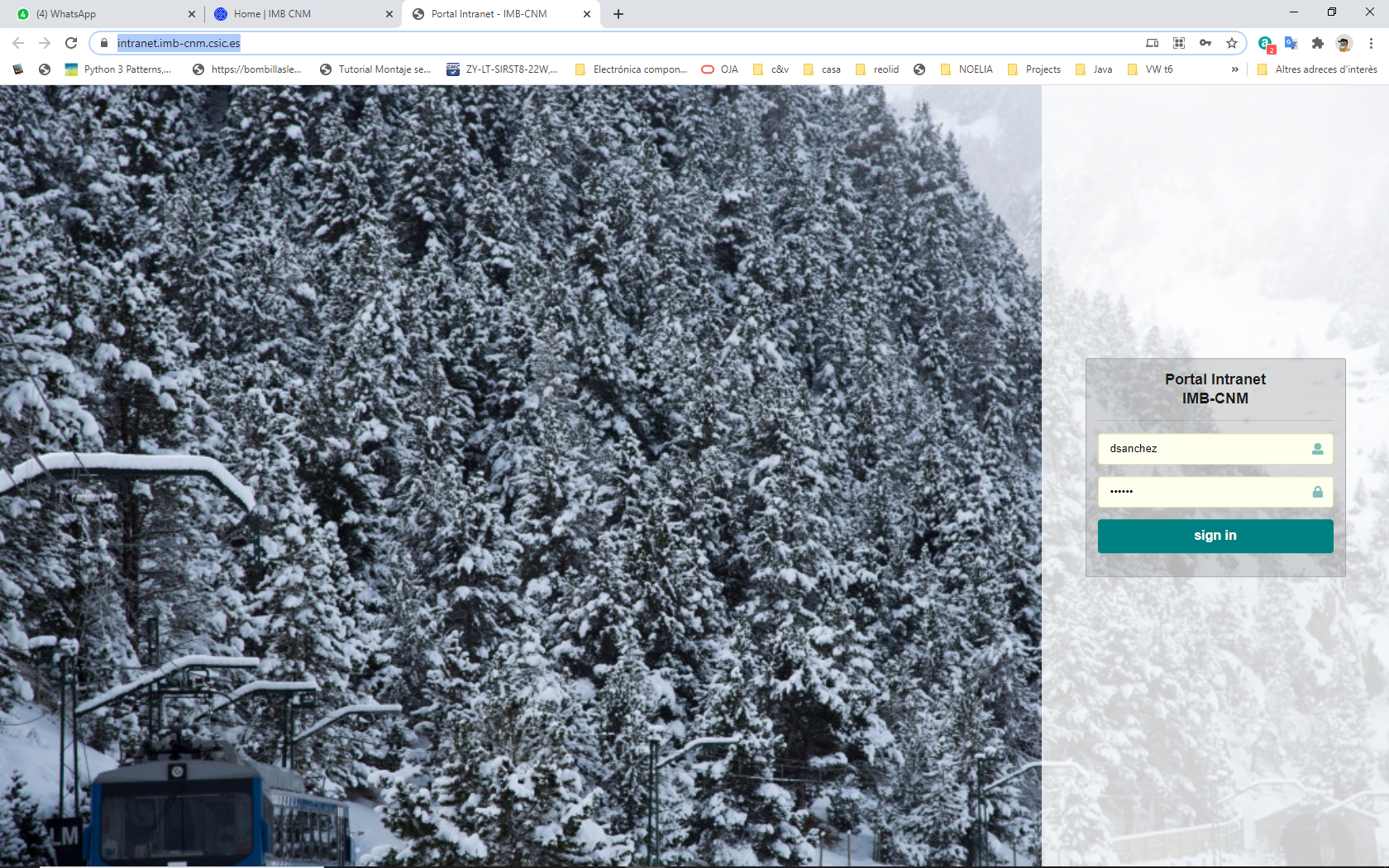
* *Posición 7*. Ultimo año, incluido, hasta el cual getCNMOrders acabará de buscar las ordenes de compra del usuario ern cuestión.
* *Posición 8*. Ruta de la carpeta donde se desea que getCNMOrders guarde los resultados de la información almacenada

# getCNMOrders paso a paso

A continuación, se contextualiza cada uno de los pasos que getCNMOrders ejecuta para la obtención de la información de las órdenes de compra.

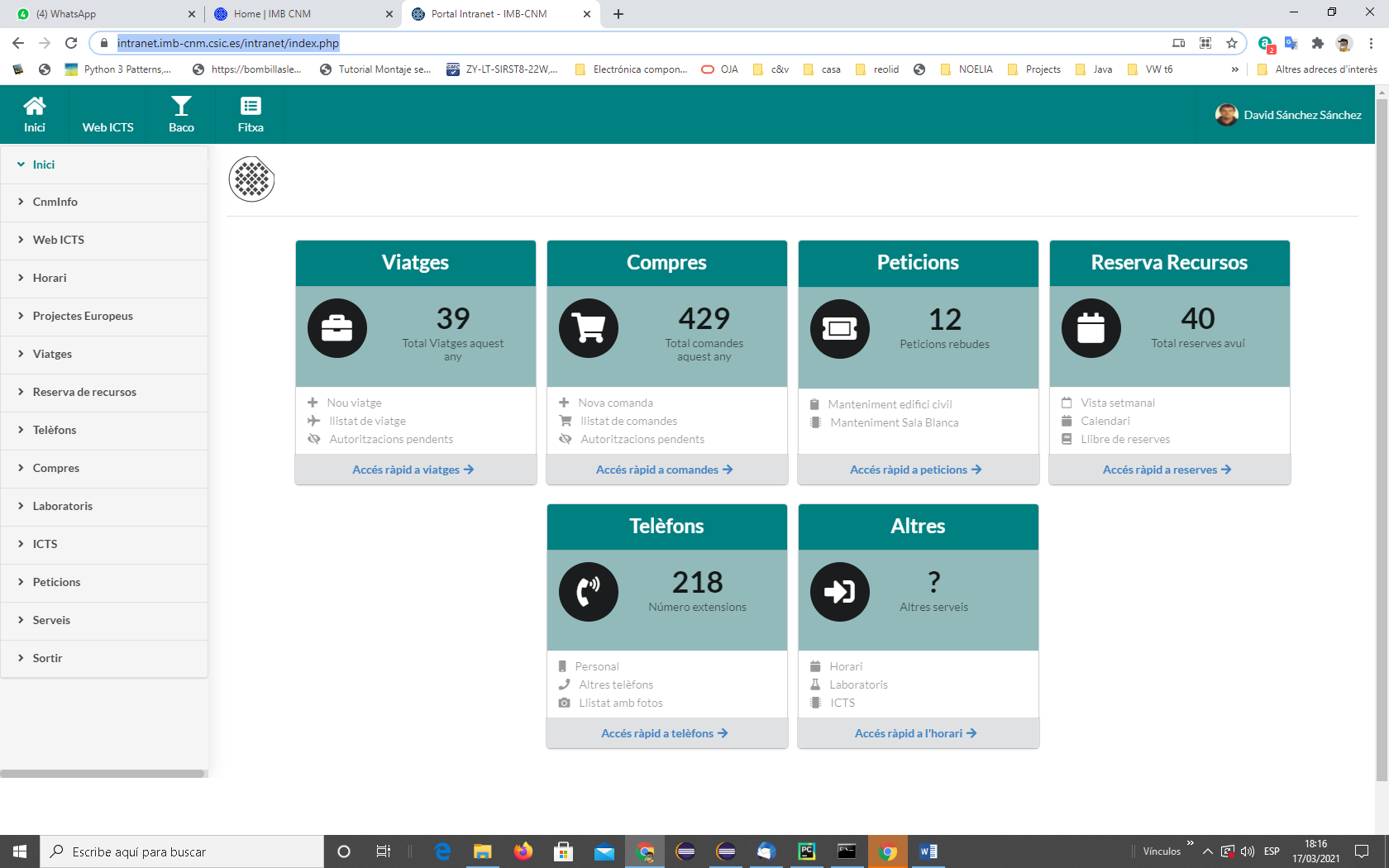
def main():  
  
 *"""  
 Clase principal que ejecuta una serie de pasos con el objetivo de acceder y guardar (serializar, json y csv) todos los datos de las ordenes de compra  
 (del imb-cnm-csic) de un periodo de tiempo marcado por dos años (parámetros de entrada) dado un usuario y pass determinado.  
 Más tarde dicho csv puede ser importado por un programa de tratamiento de datos como excel y realizar filtros (p.e. por proveedor, etc...)* ***:return****: None  
 """* #fase de recuperacion de los argumentos pasados durante la llamada al main  
 argumentsList = sys.argv[1:]  
  
 login\_url = argumentsList[0]  
 orders\_url = argumentsList[1]  
 user = argumentsList[2]  
 password = argumentsList[3]  
  
 # primer año con registros en la web de compras del cnm  
 first\_count\_year = int(argumentsList[4])  
 from\_year = int(argumentsList[5])  
  
 # correción from\_year si es menor al primer año del cual se tienen registros en la web de compras del cnm  
 if from\_year < first\_count\_year:  
 from\_year = first\_count\_year  
 to\_year = int(argumentsList[6])  
  
 # correción to\_year si es mayor al año actual  
 if to\_year > datetime.datetime.now().year:  
 to\_year = datetime.datetime.now().year  
  
 savingFilePath = argumentsList[7]  
  
 # options en caso que queramos trabajar en silent mode, etc..  
 options = Options()  
 # options.add\_argument('--headless')  
 # options.add\_argument('--disable-gpu') # Last I checked this was necessary.  
  
 driver = webdriver.Chrome(ChromeDriverManager().install(), options=options)  
  
 # fase de creación del scrapper  
 cnmWebOrdersScrapper = CNMWebOrdersScrapper(driver,  
 login\_url,  
 orders\_url,  
 user,  
 password,  
 first\_count\_year,  
 from\_year,  
 to\_year)

#antes de ejeuctar la línea de código siguiente el navegador deberá estar en dicha pagina



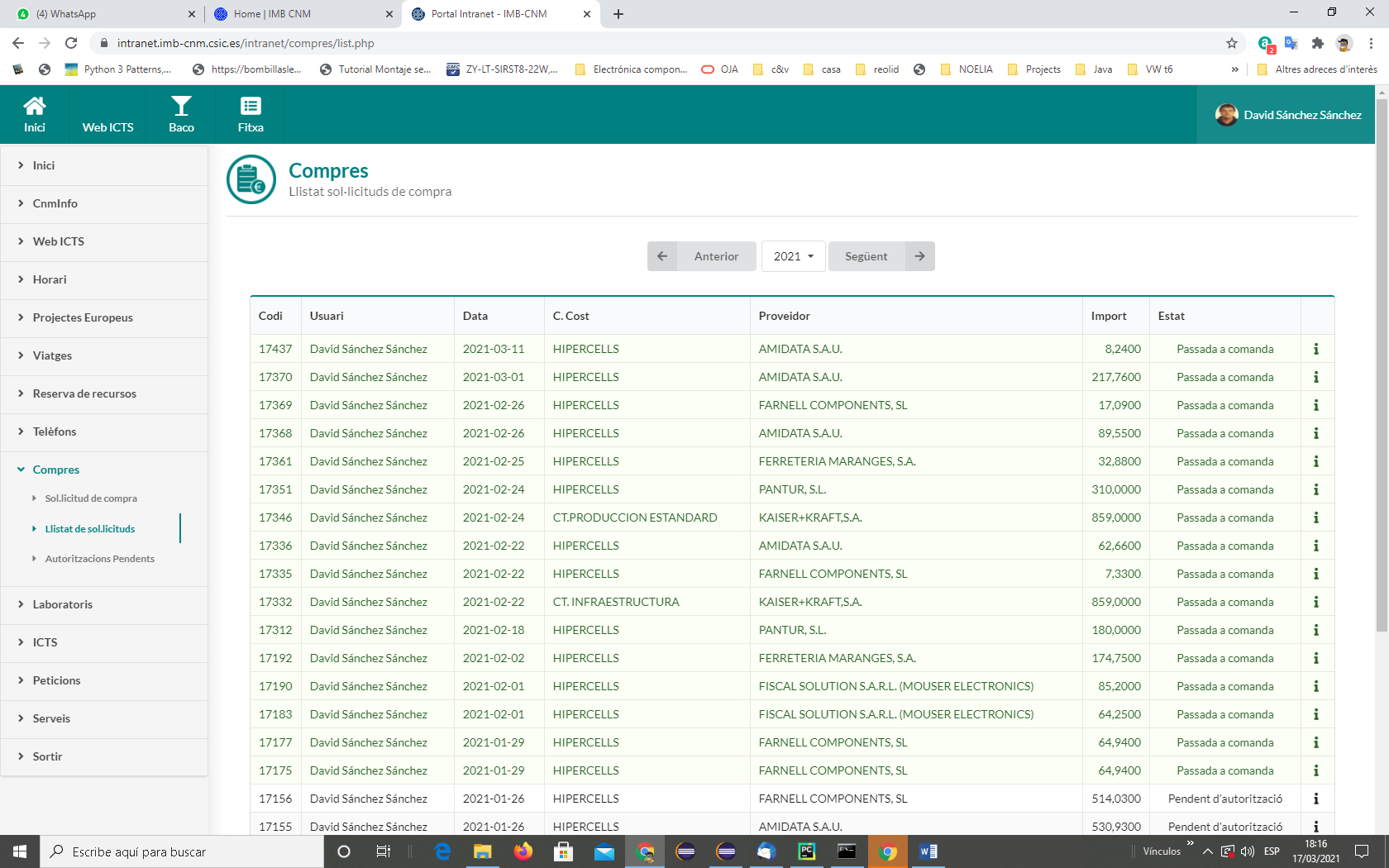
# fase de login para entrar en la intranet del cnm  
 cnmWebOrdersScrapper.doLogin()

#Despues de ejecutar la línea anterior el navegador mostrará la siguiente pagina



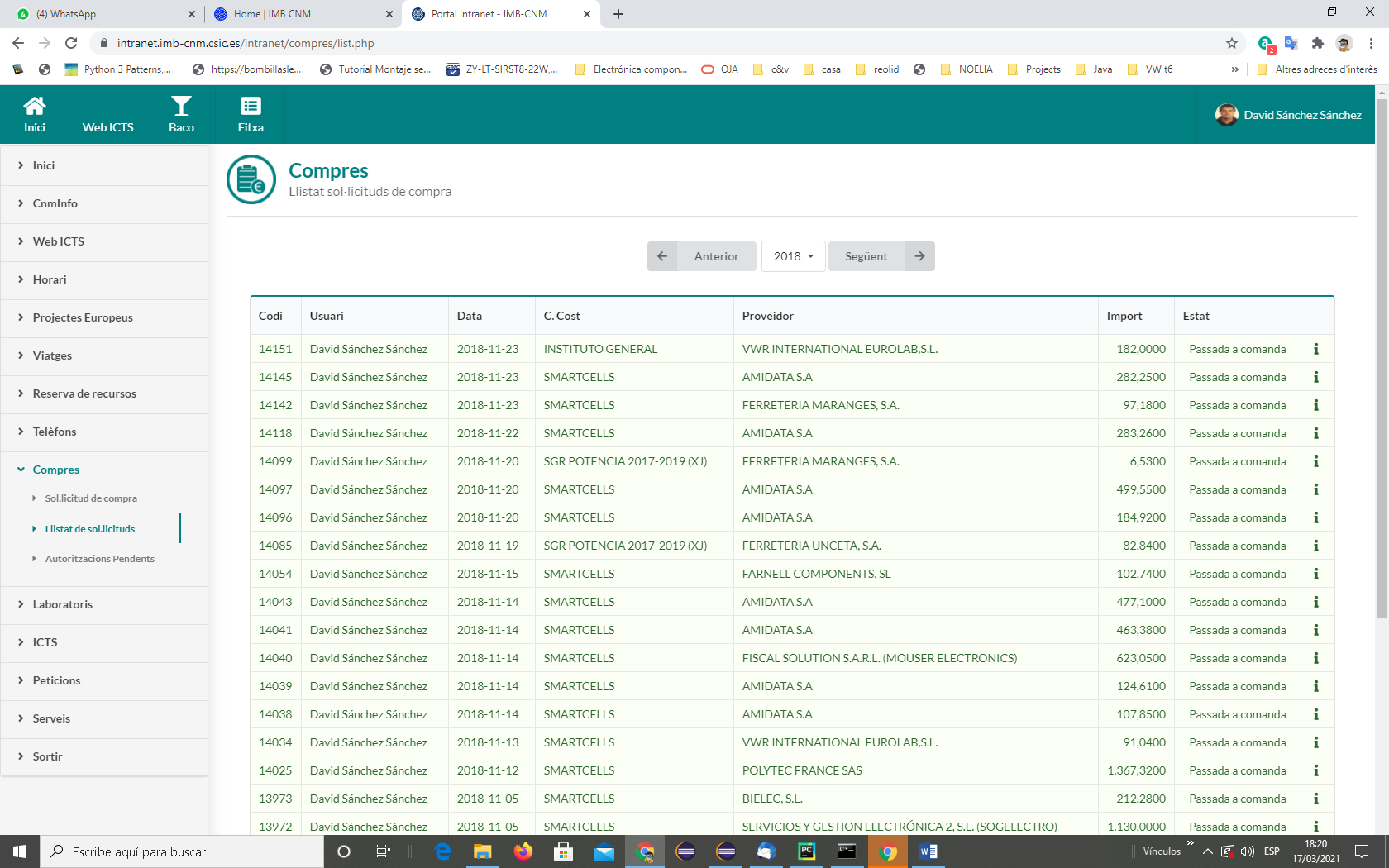
# fase de direccionamiento a una pagina determinada de la intranet del cnm  
 cnmWebOrdersScrapper.goToPage(orders\_url)

#Despues de ejecutar la línea anterior el navegador mostrará la siguiente pagina



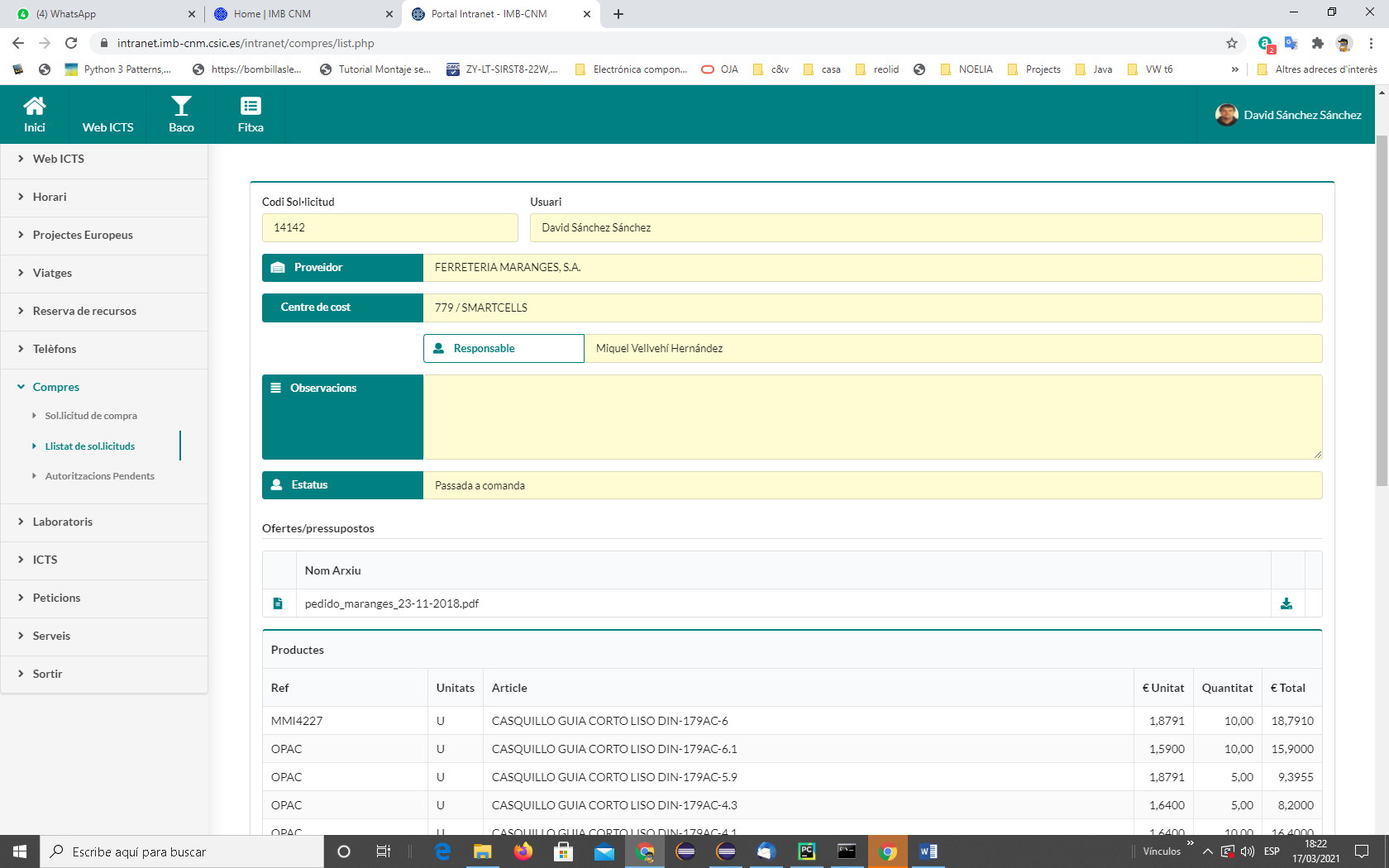
webOrders = list()  
  
 # fase de obtención de todos los datos correspondientes a los pedidos hechos y registrados en la web del cnm para el usuario dado como parámetro de entrada  
 for y in range(from\_year, to\_year + 1):  
  
 # una vez en la página del listado de solicitudes de compra de la intranet del cnm nos dirigimos al listado del año y  
 cnmWebOrdersScrapper.goToPageOfYear(y)

#Despues de ejecutar la línea anterior el navegador mostrará la siguiente pagina (por ejemplo, para y = 2018)



# espera a la carga de la pagina  
 time.sleep(1)  
 # recopilación de los códigos de los pedidos hechos en ese año y. Son necesarios para el acceso a cada una de las ordenes de compra  
 yearOrdersList = cnmWebOrdersScrapper.getActualYearPage\_OrdersList()  
  
 for code, usuario, fecha, centro\_coste, vendedor, importe, estado in yearOrdersList:  
 # acceso a la orden de compra identificada por orderCode  
 cnmWebOrdersScrapper.goToOrderPage(code)

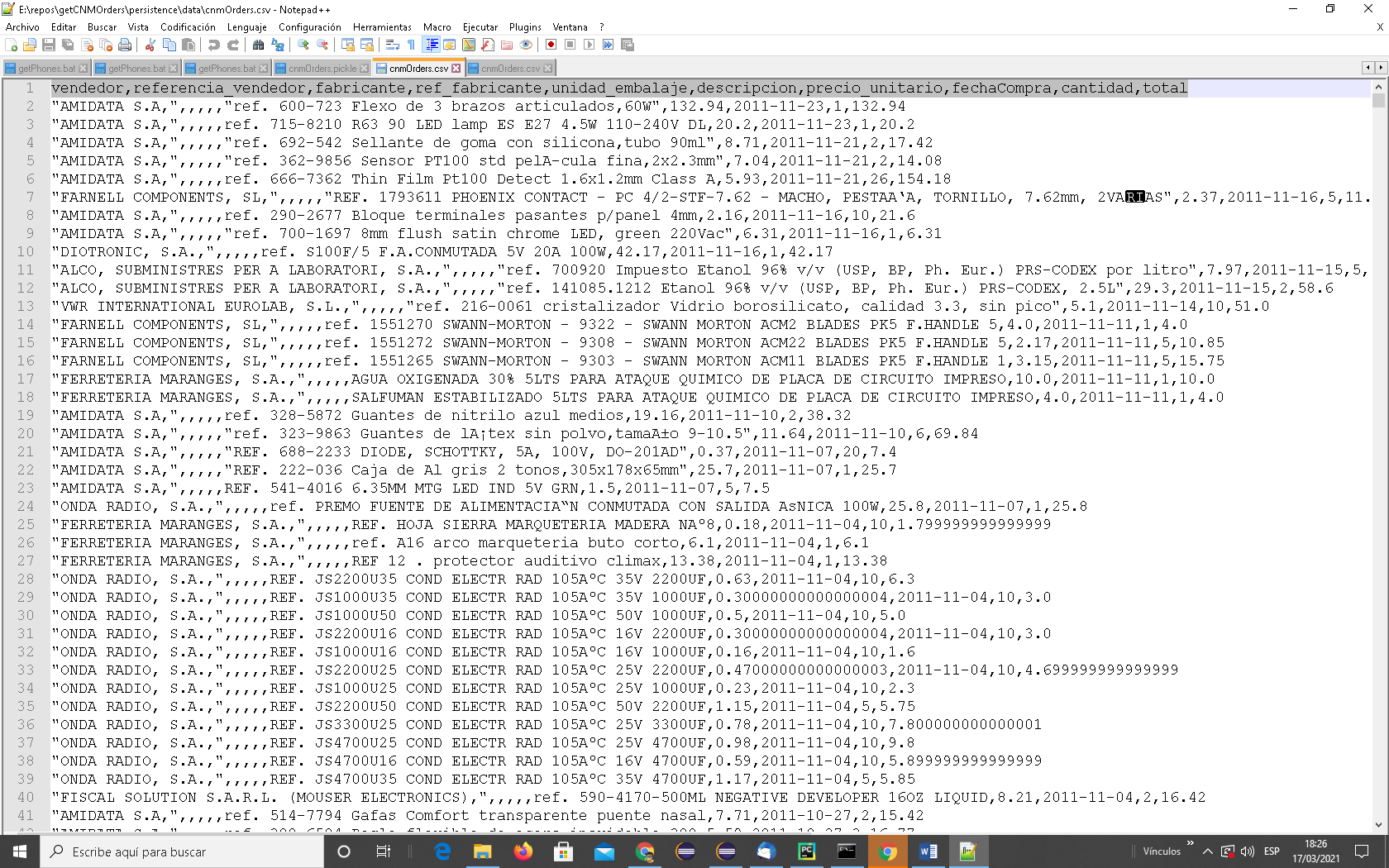
#Despues de ejecutar la línea anterior el navegador mostrará la siguiente pagina (por ejemplo para la orden de compra = 14142 del año 2018)



# espera a la carga de la pagina  
 time.sleep(2)  
 # recopilación de los datos de la orden de compra  
 webOrderData = cnmWebOrdersScrapper.getOrderData()  
 webOrderData.codigo = code  
 webOrderData.usuario = usuario  
 webOrderData.fecha = fecha  
 webOrderData.centro\_coste = centro\_coste  
 webOrderData.vendedor = vendedor  
 webOrderData.total = importe  
 webOrderData.status = estado  
  
 webOrders.append(webOrderData)  
  
 # fase de volcado de las ordenes de compra del listado webOrders a un fichero usando pickle  
 pickle\_file = open(savingFilePath + '/cnmOrders.pickle', 'wb')  
 pickle.dump(webOrders, pickle\_file)  
  
 # fase de volcado de las ordenes de compra del listado webOrders a un fichero de texto human-readable (json)  
 json\_pickle\_file = open(savingFilePath + '/cnmOrders.json', 'w')  
 json\_pickle\_file.write(jsonpickle.encode(webOrders, unpicklable=False))  
  
 # fase de volcado de las lineas de las ordenes de compra del listado webOrders a un fichero de texto human-readable (csv)  
 allWebOrders\_Lines = list()  
 for webOrder in webOrders:  
 for webOrderLine in webOrder.lineas:  
  
 try:  
 webOrderLine.product.vendedor = webOrder.vendedor  
 webOrderLine.fechaCompra = webOrder.fecha # fecha en la que se da de alta una nueva solicitud de compra  
 # en el sistema de compras de administracion del cnm  
 except KeyError:  
 # es posible que no exista historico en pedidos (orders) cuyo estado sea diferente a 'Passada a comanda'.  
 # en estos casos no hacemos nada, simplemente mostrar un mensaje por pantalla y seguir con el proceso  
 print("Pedido sin historia. No es posible acceder a la fecha del pedido")  
  
 allWebOrders\_Lines.append(webOrderLine)  
  
 #Transformación de todas las ordenes de compra a objeto json  
 wolsJSONData = json.dumps(allWebOrders\_Lines, indent=4, cls=WebOrderLine.WebOrderLineEncoder)  
 print(wolsJSONData)  
  
 #transformacion de json a df de pandas (paso intermedio con el objetivo de usar pandas para guardar las ordenes de compra en formato csv)  
 df = pd.read\_json(wolsJSONData)  
  
 #volcado de las ordenes de compra a csv  
 df.to\_csv(savingFilePath + '/cnmOrders.csv', encoding='utf-8', index=False)  
  
 # fase de finalización  
 driver.close()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

# El fichero de resultados

Todas las ordenes de compra se guardarán en un fichero tipo csv para poder importar y consultar la información obtenida facilmte con Excel.



El fichero csv, como puede verse, contendrá las siguientes columnas.

* Vendedorç
* referencia\_vendedor
* fabricante
* ref\_fabricante
* unidad\_embalaje
* descripción
* precio\_unitario
* fechaCompra
* cantidad
* total