



---

# **PROJECTE de TEORIA**

## **BASES de DADES**

---



UNIVERSITAT<sup>DE</sup>  
BARCELONA

Facultat de Matemàtiques  
i Informàtica

**Kevin Muntal Freixas**

20274516

**Yan Shchasnovich Shchasnovich**

20329536

**Joan Orteu Saiz**

20108933

**Iván Alfredo Rodrigo Arbós**

20206966

30 de Març del 2023

## 1 Introducció

Aquest projecte consisteix en dissenyar una base de dades seguint les pautes indicades pel professorat per intentar assolir els objectius determinats pel grup.

Més en concret, vam decidir que la nostra base de dades estigués centrada en el món de les finances i les matèries primeres, així com els diversos esdeveniments històrics i les parts implicades en algunes de les variacions que han patit els mercats mundials al llarg de les últimes dècades.

Vam escollir aquesta temàtica donat que sabíem que existeix multitud de fonts fiables al respecte, amb una gran quantitat de dades que podem obtenir amb facilitat. A més, la nostra base de dades podria ser útil en casos pràctics i ens proporciona una gran flexibilitat a l'hora d'establir els nostres objectius.

En la primera fase del projecte definim els objectius a assolir, mostrem el model entitat-relació de la base de dades, raonem algunes previsions sobre la futura evolució del projecte i adjuntem les fonts d'on extraurem les dades.

En la segona i última fase omplim la base de dades amb els valors trobats, els comparem entre ells amb diferents funcions d'estudi estadístic definit en codi, trobem resultats als enunciats previstos en la primera fase i obtenim les conclusions finals dels objectius amb el que vam iniciar tot aquest projecte.

## 2 Objectius

El nostre objectiu principal consisteix en intentar provar o refutar diferents enunciats. Indiquem a continuació els enunciats i una intuïció (a mode de justificació) sobre perquè pensem que són verídics o erronis:

- La fusta està correlacionada amb el petroli.
  - **Intuïció:** La fusta es talla i es transporta amb màquines que tenen un alt consum de derivats del petroli.
- El gas natural està correlacionat amb la soja.
  - **Intuïció:** No haurien de tenir cap correlació
- El preu del petroli augmenta amb les guerres (mirarem les guerres d'Ucraïna i Síria)
  - **Intuïció:** Dona la sensació que han pujat els preus després de la guerra d'Ucraïna.
- La guerra d'Ucraïna ha disminuït les seves exportacions i ha augmentat les seves importacions
  - **Intuïció:** La guerra et fa dependre d'altres països ja que la teva capacitat de producció es veu reduïda.
- La inflació està relacionada amb les guerres.
  - **Intuïció:** La inflació ve donada per l'emissió de nous diners i no directament per les guerres.
- Les sequeres augmenten el preu del suc de taronja.
  - **Intuïció:** Els tarongers necessiten molta aigua.

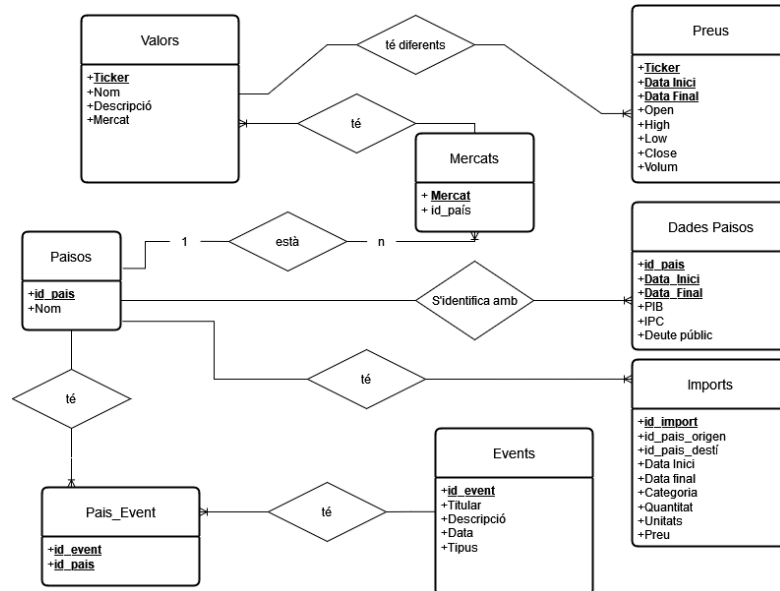
- El petroli està correlacionat amb el l'IPC global.
  - **Intuïció:** El petroli s'utilitza per tot, així que si el petroli puja, creiem que haurien de pujar els preus de tot.
- El PIB anglès està correlacionat amb el PIB Europeu.
  - **Intuïció:** Degut a que era un país Europeu, i es vol veure que després del Brexit, ja no ho està.
- Els aranzels imposats a Espanya per part de EEUU van disminuir les exportacions de pernil ibèric a EEUU i va baixar-ne el preu.
  - **Intuïció:** Haver de pagar aranzels hauria de disminuir el benefici d'exportar a EEUU. Exportar menys implica vendre menys, per tant hauria de baixar el preu.

Addicionalment, presentarem l'opció de poder escollir dos actius i veure'n la correlació dels preus al llarg d'un període donat (per exemple, or i ferro al llarg dels últims 3 anys).

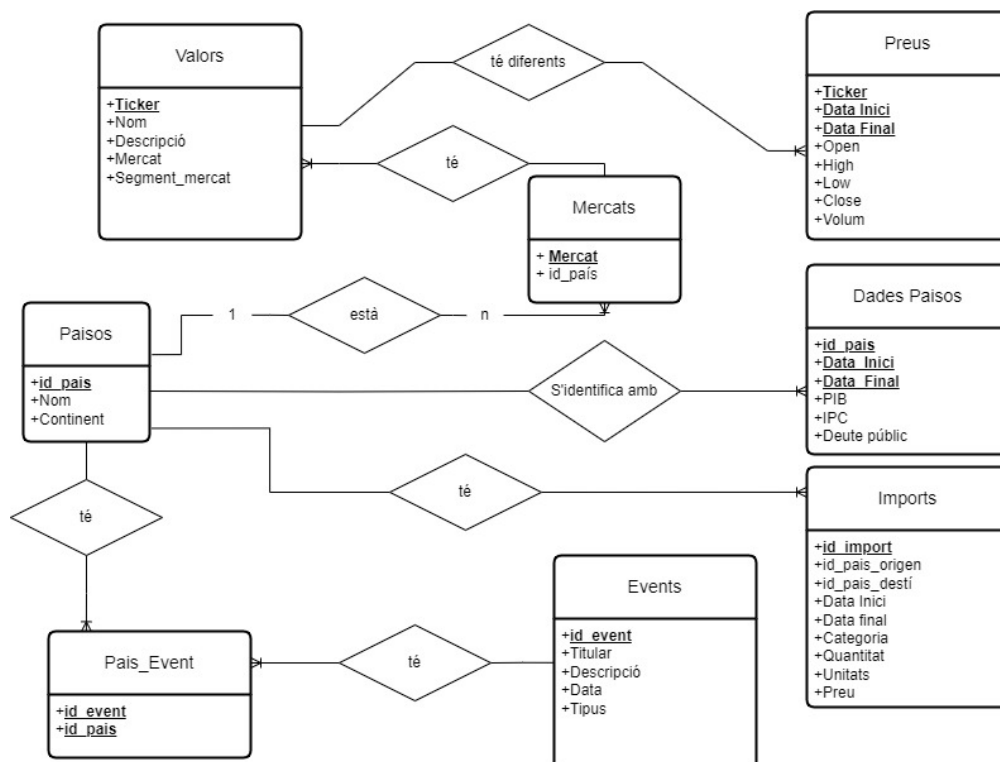
En cas de tenir més temps de l'esperat afegirem més enunciats o objectius per a la següent entrega.

### 3 Model Entitat Relació

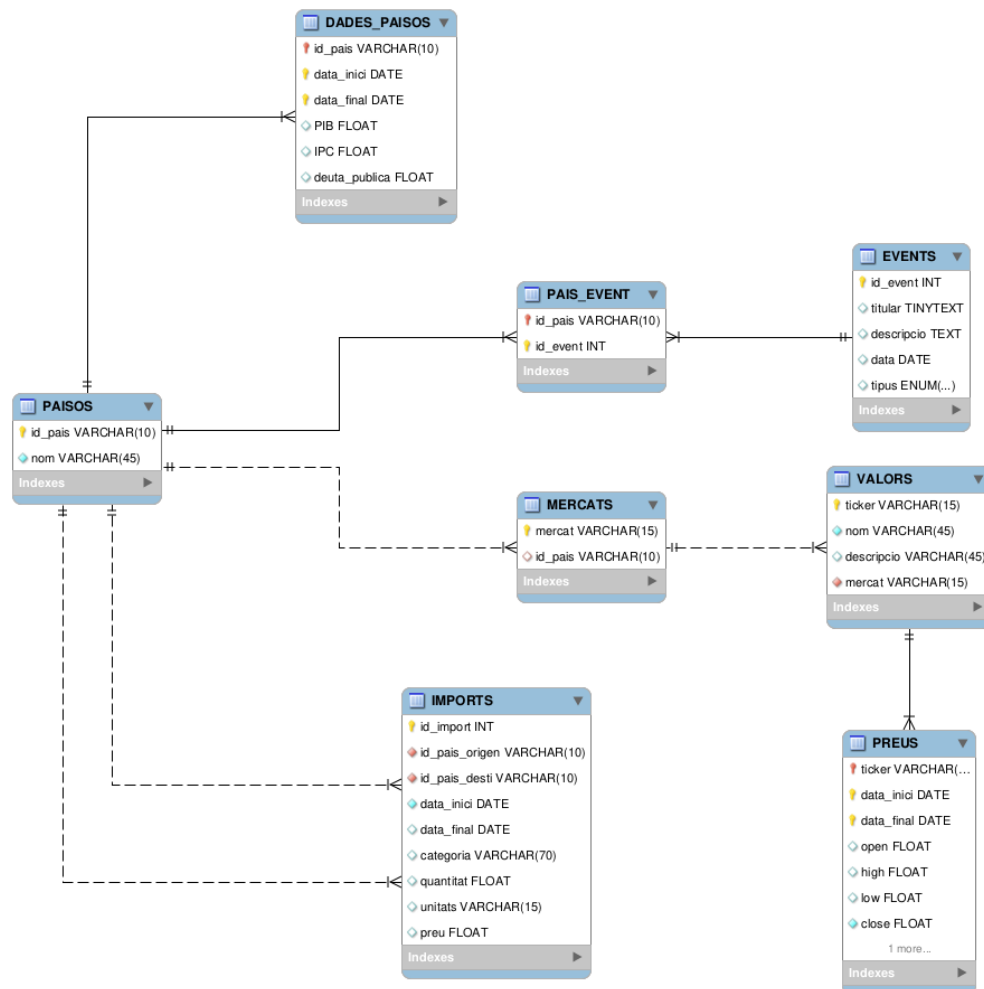
- Esquema general



- Esquema general després de plenar la bases de dades



- Model aplicat al WorkBench



### 3.1 Actualització requerida al omplir la base de dades

Inicialment teníem el camp “tipus” de la taula “events” com un *enum*, però vam decidir modificar-ne el tipus a *varchar(45)* mitjançant la comanda:

```
ALTER TABLE events
MODIFY tipus varchar(45);
```

Hem tingut la necessitat d’afegir un camp amb el continent d’un país. Hem executat:

```
ALTER TABLE paisos
ADD continent varchar(25);
```

Finalment, per actualitzar els continents, consultes de l’estil:

```
UPDATE paisos
SET
continent = 'Europa'
WHERE id_pais = "ES";
```

## 4 Origen de les dades

Recercant sobre llocs web d’on podem treure les dades que ens poden arribar a interessar, hem notat que, si bé hi ha prou disponibilitat gratuïta de dades financeres generals, en molts casos les dades més completes, amb opció d’exportació fàcil o de mercats concrets, són de pagament. Hem recopilat les fonts que ens seran més útils:

- **finance.yahoo.com:** Web d’on pots recopilar informació i preu històrics sobre diferents instruments financers. D’alguns d’ells apareix directament l’opció de descarregar CSV, i si no podem accedir a la seva **API** mitjançant enllaços amb paràmetres GET de la següent forma.
  - `https://query1.finance.yahoo.com/v7/finance/download/BRNT.L?period1=1329868800&period2=1678060800&interval=1d&events=history&includeAdjustedClose=true`
- BRNT.L és el símbol o ticker de l’actiu, i amb els paràmetres get podem especificar diferents opcions, com per exemple períodes.
- Per facilitar l’ús de l’API de yahoo finance descrita, podem utilitzar la llibreria de python **yfinance**, que en simplifica el seu ús i encapsula ja les dades en diferents formats (per exemple. en pandas dataframes).
- **nasdaq.com:** disposen d’una API gratuïta (1), així com de dades exportables (2).
  - `data.nasdaq.com/tools/api`
  - `www.nasdaq.com/market-activity/commodities`
- **Dades exportables.**
  - `www.investing.com/commodities/`

- **Preus de mercats (de ciutat) espanyols..**

- [www.mercasa.es/precios-y-mercados-mayoristas/](http://www.mercasa.es/precios-y-mercados-mayoristas/)

- **Preus i indicadors de diferents actius.**

- [www.indexmundi.com/es/precios-de-mercado](http://www.indexmundi.com/es/precios-de-mercado)

- **Preus d'aliments a Espanya. Exportable..**

- [www.mapa.gob.es/es/alimentacion/servicios/observatorio-de-precios-de-los-alimentos/default2.aspx](http://www.mapa.gob.es/es/alimentacion/servicios/observatorio-de-precios-de-los-alimentos/default2.aspx)

- **Dades del comerç macro espanyol..**

- [datacomex.comercio.es](http://datacomex.comercio.es)

- **Dades diverses de països..**

- [datosmacro.expansion.com](http://datosmacro.expansion.com)

Pel que fa a les notícies o esdeveniments, n'hi haurà prou amb recercar a qualsevol dels múltiples portals de notícies que hi ha i introduir manualment a la base de dades els que ens siguin rellevants.

Pensem que amb totes les dades que podem recopilar de les webs presentades, i d'algunes més, en tindrem suficient per satisfer múltiples objectius.

## 4.1 Obtenció i manipulació de les dades

Obtenir i manipular les dades ha representat una bona part del projecte. Encara que ja ens ho intuïem una mica, hem acabat veient que en la majoria dels casos el format de les dades no és gairebé mai exactament el que un vol o el que admet SQL. Obviant el problema d'obtenció de dades (les dades financeres de qualitat no són fàcilment obtenibles sense pagar), ens hem trobat amb varis problemes.

Per començar, necessitàvem obtenir bastantes dades sobre preus, i fer-ho manualment, evidentment, no era una bona opció. Vam utilitzar doncs el següent script de python que es connectava a la api de yahoo finance i passava els preus a un csv:

```

import yfinance as yf
import pandas as pd

# llista de tickers a consultar
tickers = ["GC=F", "SI=F", "PL=F", "HG=F", "PA=F", "CL=F", "HO=F", "NG=F", "RB=F", "BZ=F",
           "B0=F", "ZC=F", "ZO=F", "KE=F", "ZR=F", "ZM=F", "ZL=F", "ZS=F", "CC=F",
           "KC=F", "CT=F", "OJ=F", "SB=F", "LBS=F", "HE=F", "LE=F", "GF=F"]

data_list = list()
for ticker in tickers:
    data = yf.download(ticker, interval="1d", period="100y")
    data["Ticker"] = ticker
    data_list.append(data)

df = pd.concat(data_list)

df = df.reset_index()

header = ["Date", "Ticker", "Open", "High", "Low", "Close", "Volume"]

df.to_csv("./data.csv", columns = header, index=False)

```

A més, vam tenir contratemps amb el format de les dades, que hem hagut de tractar amb scripts de python i funcionalitat find and replace per tal que el format s'adeqüés al que demanava SQL. Per exemple, teníem les dades dels països en un csv amb el següent format:

```

idPais,inici,final,PIB,IPC,DeutePublic
RU,2022/03/01,2022/04/01,"0,167",
RU,2022/02/01,2022/03/01,"0,092",
RU,2022/01/01,2022/02/01,"0,087",
...

```

El problema era que les columna IPC estava envoltada de cometes (") però per importar-ho a la base de dades, on IPC estava definit com un float, aquestes cometes sobraven i calia canviar la coma dels decimals per un punt. Vam utilitzar una expressió regular per canviar-ne el format:

```

FIND: "([-]*[0-9]+),([0-9]+)"
REPLACE: $1.$2

```

En aquest cas, el find busca patrons de l'estil "-0,431" i fa dos grups, un amb un signe negatiu opcional més algun dígit (part entera del nombre), i un amb la part decimal, i canvia les coincidències per part\_entera.part\_decimal.

Ens passava també que de les columnes IPC, PIB i DeutePublic, alguns camps eren buits i MySQL diferencia entre "" i NULL, donant per tant error a l'hora d'importar-ho en aquests camps (que són floats). Abans d'importar les dades, doncs, havíem de posar en aquells camps buits la paraula NULL. Vam utilitzar les següents expressions regulars per fer-ho:



```

FIND: (,)(,)          // buscar dos comes juntes i fer dos grups
REPLACE: $1NULL$2      // posar NULL entre grups

FIND: (,)([\\n,\\s])    // busca una coma i nova linia o espai, fes dos grups
REPLACE: $1NULL$2      // canvia-ho a ,NULL\\n

```

Ens hem trobat també amb problemes a l'hora de pujar les dades a la base de dades a través de fitxers. En resum, SQL per defecte té habilitades diferents opcions de configuració que fan que pujar els fitxers, en particular amb la comanda LOAD DATA INFILE, sigui més complicat (entenem que per motius de seguretat i fiabilitat). A més, l'opció d'importar dades de csv del workbench era exageradament lenta. Vam estar provant amb diferents opcions per importar-ho, com per exemple la utilitat mysqlimport, però el final el que millor va resultar va ser afegir el csv més llarg que teníem (el de preus) mitjançant un script de python que anés fent els INSERT.

```

import mysql.connector
import csv

cnx = mysql.connector.connect(user="root", password="", database="mercatoanalysis")
cursor = cnx.cursor()
csv_data = csv.reader(open("./preus.csv"))
next(csv_data)

for row in csv_data:
    cursor.execute('INSERT INTO preus VALUES(%s,%s,%s, %s, %s, %s, %s, %s)',
        (row[2], row[0], row[1], row[3], row[4], row[5], row[6], row[7]))

cnx.commit()
cursor.close()

print("Done")

```

Notem que el pas intermedi de passar-ho a csv va ser un pas redundant, i que podríem haver passat directament les dades des de la api a la base de dades. Segurament hagués set millor, però d'aquesta manera amb csv “intermedi” també podríem comprovar quines eren les dades que realment inseríem a la base de dades abans de fer-ho i mirar que el format fos correcte.

## 5 Implementació de la base de dades

### 5.1 Països

Tenim tres graelles dedicades als països:

idPaís	nom
AF	Afganistán
AL	Albania
DE	Alemania
AD	Andorra
FM	Estados Federados de Micronesia
US	Estados Unidos de América
EE	Estonia
SZ	Esuatini
ET	Etiopía
YE	Yemen
DJ	Yibuti
ZM	Zambia
ZW	Zimbabue

Taula 1: Relació de tots els identificadors amb el país (hi ha tots els països, aquesta és una petita mostra).

idPaís	nom
US	Estats Units
ES	Espanya
GB	Regne Unit
CA	Canada
AE	Emirats Arabs Units
VE	Venezuela
GER	Alemania
SI	Siria
UKR	Ucrania
FR	Francia
RU	Russia

Taula 2: Països que volem estudiar i l'identificador que li correspon a cadascun.

idPais	inici	final	PIB	IPC	DeutePublic
RU	2022/03/01	2022/04/01	NULL	0.167	NULL
RU	2022/02/01	2022/03/01	NULL	0.092	NULL
RU	2022/01/01	2022/02/01	NULL	0.087	NULL
RU	2021/12/01	2022/01/01	NULL	0.084	255354
ES	2022/11/01	2022/12/01	7255	0.068	1502543
ES	2022/10/01	2022/11/01	7255	0.073	1502543
ES	2022/09/01	2022/10/01	7018	0.089	1503760
ES	2022/08/01	2022/09/01	7018	0.015	1503760
ES	2022/07/01	2022/08/01	7018	0.108	1503760
VE	2020/03/01	2020/04/01	NULL	24.306	NULL
VE	2020/02/01	2020/03/01	NULL	29.109	NULL
VE	2020/01/01	2020/02/01	NULL	51.97	NULL
VE	2019/12/01	2020/01/01	NULL	95.855	NULL

Taula 3: Els PIB, IPC, Deute Pública i les dates del valor del país a estudiar. S'indiquen amb l'identificador corresponent (hi ha moltes més dades, aquesta és una petita mostra).

## 5.2 Mercats

Tenim una petita taula amb els noms dels mercats i el país en el que estan situats.

mercat	idPais
NYMEX	US
CME	US
CBOT	US
ICE	US
COMEX	US
CME	US

Taula 4: Els identificadors dels mercats amb el país corresponent.

### 5.3 Valors

Una única taula per ampliar les dades dels mercats.

Ticker	Nom	Descripció	Mercat	Segment de Mercat	
GC=F	Gold		COMEX	Metal	
SI=F	Silver		COMEX	Metal	
PL=F	Platinum Jul 23		NYMEX	Metal	
HG=F	Copper Jul 23		COMEX	Metal	
PA=F	Palladium Jun 23		NYMEX	Metal	
CL=F	Crude Oil		NYMEX	Energy	
HO=F	Heating Oil Jun 23		NYMEX	Energy	
NG=F	Natural Gas Jun 23		NYMEX	Energy	
RB=F	RBOB Gasoline Jun 23		NYMEX	Energy	
BZ=F	Brent Crude Oil Last Day Financ		NYMEX	Energy	
B0=F	Mont Belvieu LDH Propane (OPIS)		NYMEX	Energy	
ZC=F	Corn Futures	Jul-2023		CBOT	Agriculture
ZO=F	Oat Futures	Jul-2023		CBOT	Agriculture
KE=F	KC HRW Wheat Futures	Jul-2023		CBOT	Agriculture
ZR=F	Rough Rice Futures	Jul-2023		CBOT	Agriculture
ZM=F	Soybean Meal Futures	Jul-2023		CBOT	Agriculture
ZL=F	Soybean Oil Futures		CBOT	Agriculture	
CC=F	Cocoa Jul 23		ICE	Agriculture	
KC=F	Coffee Jul 23		ICE	Agriculture	
CT=F	Cotton Jul 23		ICE	Agriculture	
OJ=F	Orange Juice Jul 23		ICE	Agriculture	
SB=F	Sugar11 Jul 23		ICE	Agriculture	
LBS=F	Lumber May 23		CME	Wood	
HE=F	Lean Hogs Futures	Jun-2023		CME	Livestock
LE=F	Live Cattle Futures	Jun-2023		CME	Livestock
GF=F	Feeder Cattle Futures	Aug-2023		CME	Livestock

Taula 5: Valors dels mercats que ens ajuda a classificar-los i introduir els preus.

## 5.4 Preus

La taula més gran de tota la base, en aquesta s'emmagatzemen preus de tots mercats i tota classe de productes.

Date	dateFinal	Ticker	Open	High	Low	Close	Volume
2000-08-30	2000-08-31	GC=F	273.8999938964844	273.8999938964844	273.8999938964844	273.8999938964844	0
2000-08-31	2000-09-01	GC=F	274.79998779296875	278.29998779296875	274.79998779296875	278.29998779296875	0
2000-09-01	2000-09-02	GC=F	277.0	277.0	277.0	277.0	0
2000-09-05	2000-09-06	GC=F	275.79998779296875	275.79998779296875	275.79998779296875	275.79998779296875	2
2005-04-13	2005-04-14	PI=F	861.0	861.0	861.0	867.5999755859375	15
2005-04-14	2005-04-15	PI=F	862.0	862.0	862.0	861.9000244140625	1
2005-04-15	2005-04-16	PI=F	859.0	859.0	859.0	864.5999755859375	4
2005-04-18	2005-04-19	PI=F	864.5	864.5	864.5	864.5	30
2005-04-19	2005-04-20	PI=F	868.0	870.0	868.0	873.2000122070312	5
2005-04-20	2005-04-21	PI=F	869.0	869.0	869.0	879.5999755859375	12
2005-04-21	2005-04-22	PI=F	876.4000244140625	876.4000244140625	876.4000244140625	876.4000244140625	12
2005-04-22	2005-04-23	PI=F	882.0	882.0	880.7999877929688	878.5	34
2004-03-01	2004-03-02	KE=F	384.0	385.0	376.75	379.5	659
2004-03-02	2004-03-03	KE=F	377.5	381.0	376.0	379.0	181
2004-03-03	2004-03-04	KE=F	377.0	377.0	373.0	373.0	103
2004-03-04	2004-03-05	KE=F	372.0	375.0	371.0	375.0	59
2004-03-05	2004-03-06	KE=F	374.0	375.0	371.5	372.25	96
2004-03-08	2004-03-09	KE=F	368.0	373.0	367.0	369.0	26
2004-03-09	2004-03-10	KE=F	369.0	369.0	369.0	369.0	4
2004-03-10	2004-03-11	KE=F	369.0	370.0	362.0	365.0	7
2023-05-04	2023-05-05	GF=F	203.6999969482422	204.5	203.4250030517578	203.64999389648438	1676
2023-05-05	2023-05-06	GF=F	203.64999389648438	204.5	202.1750030517578	202.52499389648438	3084
2023-05-08	2023-05-09	GF=F	202.1750030517578	203875	202.0	203.4250030517578	3084
2023-05-09	2023-05-10	GF=F	224.0500030517578	226.5	223125	225.9499969482422	6923

## 5.5 Events

Una primera taula general per relacionar els esdeveniments amb els països i una altra que identifica cada event a estudiar.

idEvent	Titular	Descripció	Data	Tipus
1	Guerra esclata a Syria	En un país amb geolocalització clau per desestabilitzar el preu del petroli i gas.	2011-03-15	Guerra
2	Guerra esclata a Ucraïna	En un país amb geolocalització clau per desestabilitzar el preu del petroli i el gas.	2022-02-24	Guerra
3	La gran sequera del 2007	Està sent un any de gran sequies.	2007-08-08	Meteorològic
4	El brexit és una realitat	UK surt de la Unió Europea	2020-01-31	Polític

Taula 6: Events importants a destacar

idEvent	idPaís
1	DE
1	US
1	RU
1	ES
1	FR
1	IT
1	GB
2	DE
2	US
2	RU
2	ES
2	FR
2	IT
2	GB
3	ES
4	DE
4	US
4	RU
4	ES
4	FR
4	IT
4	GB

Taula 7: Relacionar l'event amb el país a partir dels identificadors.

## 5.6 Imports

Finalment, una ultima graella amb els valors de les importacions.

id_import	id_pais_origen	id_pais_desti	data_inici	data_final	categoria	quantitat	unitats	preu
1	ES	DE	01-01-2023	31-12-2023	Conservas de carne	3559486,51	kg	24708602,49
1	ES	DE	01-01-2022	31-12-2022	Conservas de carne	15987027,71	kg	104091147
1	ES	DE	01-01-2021	31-12-2021	Conservas de carne	18071797,82	kg	104577093,9
1	ES	DE	01-01-2020	31-12-2020	Conservas de carne	16514537,19	kg	95390225,63
1	ES	DE	01-01-2019	31-12-2019	Conservas de carne	16755924,46	kg	86858423,42
1	ES	DE	01-01-2018	31-12-2018	Conservas de carne	19333637,68	kg	97305224,03
1	ES	DE	01-01-2017	31-12-2017	Conservas de carne	15585465,69	kg	88145436,74
1	ES	DE	01-01-2016	31-12-2016	Conservas de carne	12085930,8	kg	71416985,78
1	ES	DE	01-01-2015	31-12-2015	Conservas de carne	15688109,75	kg	84734780,54
1	ES	DE	01-01-2014	31-12-2014	Conservas de carne	14784665,99	kg	73735458,8
1	ES	DE	01-01-2013	31-12-2013	Conservas de carne	13771935,45	kg	72343747,68
1	ES	DE	01-01-2012	31-12-2012	Conservas de carne	13812719,69	kg	72915587,96
1	ES	DE	01-01-2011	31-12-2011	Conservas de carne	10050951,25	kg	71856543,65
1	ES	DE	01-01-2010	31-12-2010	Conservas de carne	9098587,82	kg	55097181,43
1	ES	DE	01-01-2009	31-12-2009	Conservas de carne	11679665,99	kg	43902416,49
1	ES	DE	01-01-2008	31-12-2008	Conservas de carne	10252133,33	kg	43777343,17
1	ES	DE	01-01-2007	31-12-2007	Conservas de carne	10203484	kg	43501165,92
1	ES	DE	01-01-2006	31-12-2006	Conservas de carne	9573622,62	kg	42750892,53
1	ES	DE	01-01-2005	31-12-2005	Conservas de carne	9879987,29	kg	42605941,66
1	ES	DE	01-01-2004	31-12-2004	Conservas de carne	8345940,55	kg	33081695,24
1	ES	DE	01-01-2003	31-12-2003	Conservas de carne	9336030,26	kg	32396438,81
1	ES	DE	01-01-2002	31-12-2002	Conservas de carne	11230053,71	kg	38821533,13
1	ES	DE	01-01-2001	31-12-2001	Conservas de carne	16738759,4	kg	57200171,79
1	ES	DE	01-01-2000	31-12-2000	Conservas de carne	9905581,28	kg	31229285,42

Taula 8: Importacions amb les seves dades per cada país

## 7 Estudi estadístic

En les següents pàgines veurem l'evolució de tot el codi per crear les funcions que ens ajuden a respondre els enunciats descrits en la primera part del document. Entre aquestes anem argumentant els resultats obtinguts i relacionant-los amb la intuïció que teníem abans de fer aquest estudi.

```
[1]: password = ""
      database= "mercatoanalysis"
```

```
[2]: import mysql.connector
      import datetime
      import pandas as pd
      import matplotlib.pyplot as plt
      import mplfinance as mpf
      import numpy as np
      import ta
```

```
[3]: def getPriceData(ticker, initial_date, period, db):
      ini_date = datetime.datetime.strptime(initial_date, "%d-%m-%Y")

      end_date = ini_date + datetime.timedelta(days=period)

      # query = 'SELECT {} FROM {} WHERE ticker=%s AND data_inici>=%s AND
      ↪data_inici < %s'.format(columns, "preus") # string formatting method

      db.execute("SELECT * FROM preus "
                  "WHERE ticker=%s AND data_inici>=%s AND data_inici <
      ↪%s", (ticker, ini_date, end_date))

      result = db.fetchall()
      df = pd.DataFrame(result, columns=('ticker', 'data_inici',
      ↪'data_final', 'open', 'high', 'low', 'close', 'volume'))
      df['data_inici'] = pd.to_datetime(df['data_inici']) # set dates to
      ↪datetime64
      df['data_final'] = pd.to_datetime(df['data_final'])
      df.set_index('data_inici', inplace=True)
      return df
```



```
[4]: def getCountryData(country_code, initial_date, period, db,
    ↪columns=None):
    ini_date = datetime.datetime.strptime(initial_date, "%d/%m/%Y")

    end_date = ini_date + datetime.timedelta(days=period)

    db.execute("SELECT * FROM dades_paisos WHERE "
        ↪"id_pais=%s AND data_inici>=%s AND data_inici < %s",
    ↪(country_code, ini_date, end_date))

    result = db.fetchall()
    df = pd.DataFrame(result, columns=('id_pais', 'data_inici',
    ↪'data_final', 'PIB', 'IPC', 'deuta_publica'))
    df['data_inici'] = pd.to_datetime(df['data_inici']) # set dates to
    ↪datetime64
    df['data_final'] = pd.to_datetime(df['data_final'])
    df.set_index('data_inici', inplace=True)

    return df
```

```
[5]: def getWorldIPC(initial_date, period, db, columns=None):
    ini_date = datetime.datetime.strptime(initial_date, "%d/%m/%Y")

    end_date = ini_date + datetime.timedelta(days=period)

    db.execute("SELECT data_inici,avg(IPC) FROM dades_paisos "
        ↪"WHERE id_pais IN ('US', 'GB', 'ES', 'CA', 'RU', 'DE') "
        ↪"GROUP BY data_inici "
        ↪"HAVING data_inici>=%s AND data_inici < %s", (ini_date,
    ↪end_date))

    result = db.fetchall()
    df = pd.DataFrame(result, columns=('data_inici', 'ipc_avg'))
    df['data_inici'] = pd.to_datetime(df['data_inici']) # set dates to
    ↪datetime64
    df.set_index('data_inici', inplace=True)

    return df
```

```
[6]: def getEventData(partial_title, db):
    title = '%' + partial_title + '%'

    db.execute("SELECT * FROM events "
               "WHERE titular LIKE %s", (title,))

    result = db.fetchall()
    df = pd.DataFrame(result, columns=('id_event', 'titular', 'descripcio', 'data', 'tipus'))
    df['data'] = pd.to_datetime(df['data']) # set dates to datetime64
    df.set_index('id_event', inplace=True)

    return df

[7]: def getImportsData(pais_origen, pais_desti, partial_categoria, db):
    categoria = '%' + partial_categoria + '%'
    db.execute('SELECT * FROM imports WHERE id_pais_origen=%s AND id_pais_desti=%s AND categoria LIKE %s',
               (pais_origen, pais_desti, categoria))

    result = db.fetchall()
    df = pd.DataFrame(result, columns=('id_import', 'id_pais_origen', 'id_pais_desti', 'data_inici', 'data_final', 'categoria', 'quantitat', 'unitats', 'preu'))
    df['data_inici'] = pd.to_datetime(df['data_inici']) # set dates to datetime64
    df['data_final'] = pd.to_datetime(df['data_final'])
    df.set_index('data_inici', inplace=True)

    return df

[8]: from warnings import simplefilter

    # To ignore warnings
    np.seterr(divide='ignore', invalid='ignore')
    simplefilter(action='ignore', category=FutureWarning)
```

## 7.1 Correlacions

Per demostrar les diferents hipòtesis de correlacions, s'han fet dos funcions, una per calcular la matriu de correlacions entre diferents valors (`calculate_corr_matrix`) i l'altre per mostrar els resultats en forma de visual amb colors (`plot_correlations`). Es calculen les taules de correlacions a través de la funcions de Pandas "corr", i s'usen els mètodes de Pearson i Spearman. En aquestes correlacions un resultat de 1, significa completament correlacionats, 0 és completament incorrelacionats i -1 és que tenen una correlació inversa.

```
[9]: import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# args = [nom_1,nom_2,...,nom_n],data1,data2,..., datan

def calculate_corr_matrix(*args):
    # Merge the dataframes based on the common identifier
    merged_df = pd.merge(args[1], args[2], on='data_inici')

    merged_df = merged_df.rename(columns={merged_df.columns[0]:
    ↪args[0][0]})
    merged_df = merged_df.rename(columns={merged_df.columns[1]:
    ↪args[0][1]})

    # Merge additional dataframes and rename columns
    for i in range(3, len(args)):
        merged_df = pd.merge(merged_df, args[i], on='data_inici')
        merged_df = merged_df.rename(columns={merged_df.columns[i-1]:
    ↪args[0][i-1]})

    # Calculate Pearson correlation matrix
    pearson_matrix = merged_df.corr(method='pearson')

    # Calculate Spearman correlation matrix
    spearman_matrix = merged_df.corr(method='spearman')
    #print(pearson_matrix)
    #print(spearman_matrix)

    return [pearson_matrix,spearman_matrix]
```

```
[10]: def plot_correlations(corr_matrix, tipus):
    mask = np.triu(np.ones_like(corr_matrix, dtype=bool))

    # Set up the figure and axes
    fig, ax = plt.subplots(figsize=(8, 6))

    # Create the heatmap using seaborn
    sns.heatmap(corr_matrix, annot=True, mask=mask, cmap='coolwarm',
                fmt='.2f', linewidths=0.5, ax=ax, vmin=-1, vmax=1)

    # Customize the plot
    ax.set_title('Matriu de correlacions de ' + tipus)
    ax.set_xticklabels(corr_matrix.columns, rotation=45)
    ax.set_yticklabels(corr_matrix.columns, rotation=0)
    plt.show()
```

### 7.1.1 Correlacions de diverses commodities

Ara calcularem les correlacions entre el petroli, el gas, la fusta i la soja, que ens serbiran per comprovar dos de les hipòtesis a mostrar: - La fusta està correlacionada amb el petroli. - El gas natural està correlacionat amb la soja.

```
[11]: #Agafem les dades de la base de dades

cnx = mysql.connector.connect(user="root", password=password,
                               database=database)
cursor = cnx.cursor()

oil = getPriceData('CL=F', "01-01-2011", 3000, cursor)['close']
gas = getPriceData('NG=F', "01-01-2011", 3000, cursor)['close']
fusta = getPriceData('LBS=F', "01-01-2011", 3000, cursor)['close']
soja = getPriceData('ZM=F', "01-01-2011", 3000, cursor)['close']

cnx.commit()
cursor.close()

#calculem les matrius

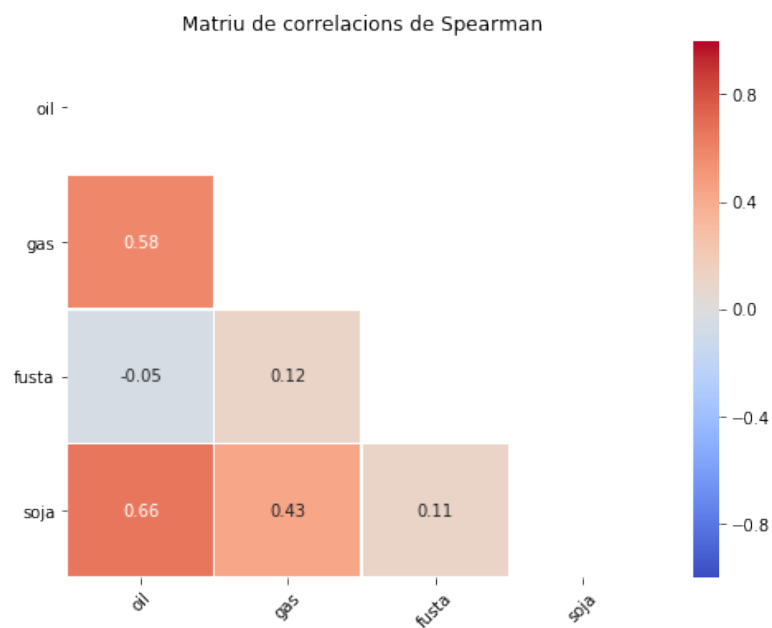
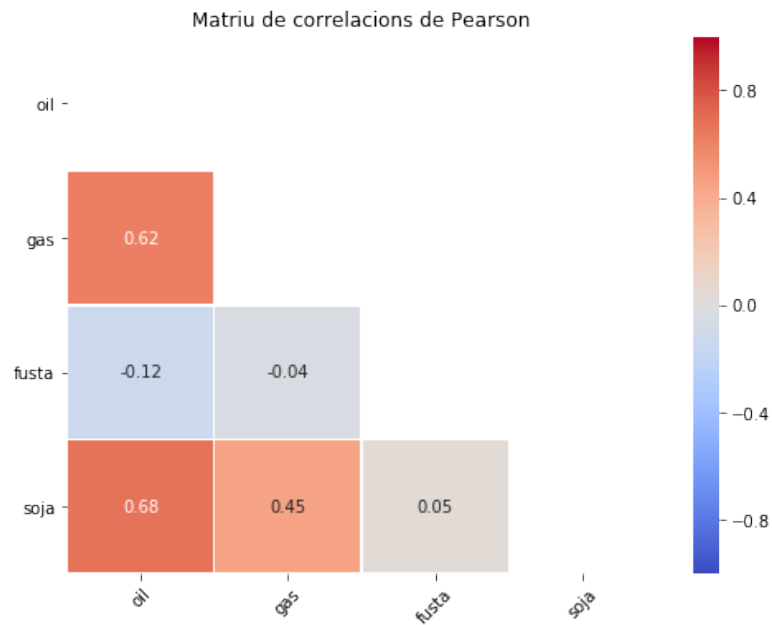
vector_corr_matrix=calculate_corr_matrix(["oil","gas","fusta","soja"],
```

```
oil,gas,fusta,soja)
```

```
#mostrem els resultats
```

```
plot_correlations(vector_corr_matrix[0], "Pearson")
```

```
plot_correlations(vector_corr_matrix[1], "Spearman")
```



En els gràfics mostrats, es pot observar més informació que la que necessitem per a comprovar les nostres hipòtesis, anem a centrar-nos en la informació rellevant: - La fusta està correlacionada amb el petroli. És una hipòtesis errònia segons els resultats obtinguts, ja que la correlació entre la fusta i el petroli segons Pearson és de -0.12 i segons Spearman és de -0.05, que són valors molt pròxims a 0 i per tant, no estan correlacionats. - El gas natural està correlacionat amb la soja. Amb les dades podem dir que estan força correlacionats, ja que s'han obtingut valors de correlació de 0.45 i 0.43 de Pearson i Spearman respectivament.

### 7.1.2 Correlació entre PIB anglès post brexit amb el PIB Europeu

Ara calcularem la correlació entre el PIB anglès i el PIB Europeu que ens serviran per comprovar la següent hipòtesis: - El PIB anglès no està correlacionat amb el PIB Europeu post brexit

```
[12]: def getEuropeanCountriesData(initial_date, period, db):
    ini_date = datetime.datetime.strptime(initial_date, "%d/%m/%Y")

    end_date = ini_date + datetime.timedelta(days=period)

    db.execute("SELECT d.id_pais,d.data_inici,d.data_final,d.PIB,d.
    ↳IPC,d.deuta_publica FROM "
               "dades_paisos d JOIN paisos p ON d.id_pais = p.id_pais "
               "WHERE p.continent = 'Europa' AND data_inici>=%s AND
    ↳data_inici<=%s ", (ini_date, end_date))

    result = db.fetchall()
    df = pd.DataFrame(result, columns=('id_pais', 'data_inici',
    ↳'data_final', 'PIB', 'IPC', 'deuta_publica'))
    return df
```

```
[13]: cnx = mysql.connector.connect(user="root", password=password,
    ↳database=database)
    cursor = cnx.cursor()

    df = getEuropeanCountriesData("01/01/2020", 999, cursor)
    pib_europeu = df[["data_inici","PIB"]]
    pib_europeu = pib_europeu.groupby("data_inici").sum()['PIB']
    pib_europeu.index = pd.to_datetime(pib_europeu.index)
```

```

PIB_Great_britain = getCountryData("GB", "01/01/2020", 999,
    cursor)['PIB']

cnx.commit()
cursor.close()

#calculem les matrius

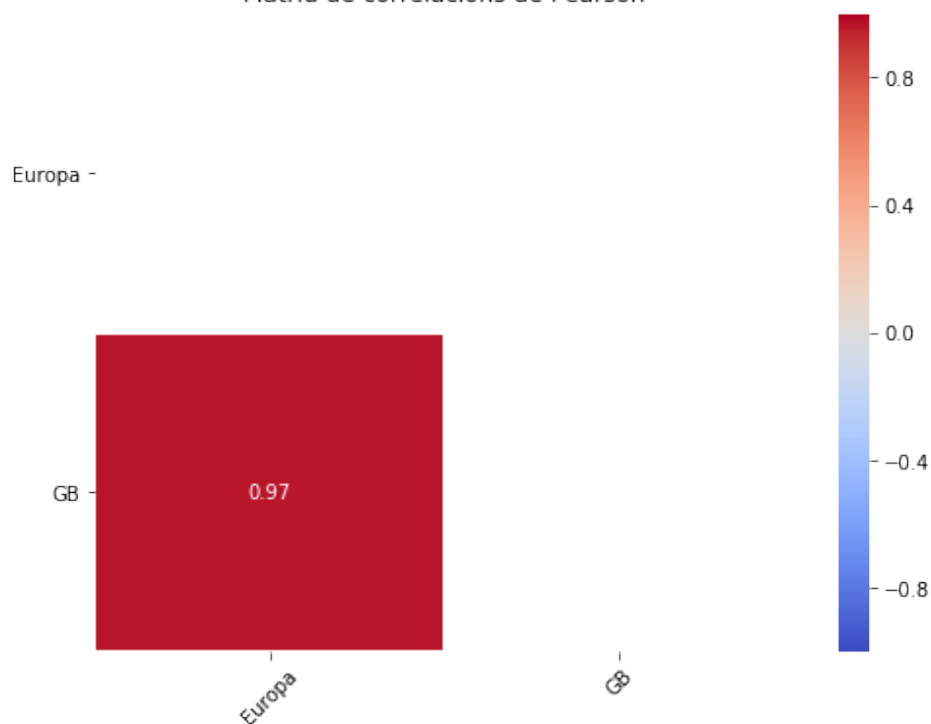
vector_corr_matrix =
    calculate_corr_matrix(["Europa", "GB"], pib_europeu, PIB_Great_britain)

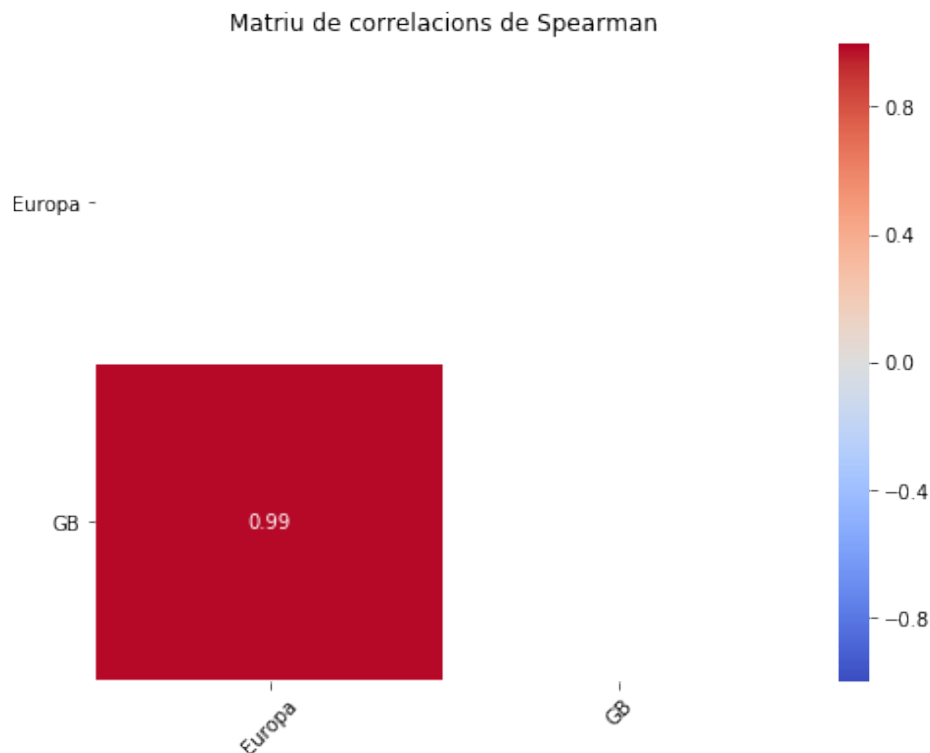
#mostrem els resultats

plot_correlations(vector_corr_matrix[0], "Pearson")
plot_correlations(vector_corr_matrix[1], "Spearman")

```

Matriu de correlacions de Pearson





El resultat és molt clar, el PIB anglès i l'Europeu, estan estretament coorelacionats amb uns coeficients de corelació de 0.97 i 0.99 de pearson i spearman respectivament. Per tant, podem refutar la hipòtesis, ja que veiem que si que estan estretament relacionats.

### 7.1.3 Correlció entre preu del petroli i IPC global

Ara calcularem la correlació entre el petroli i l'IPC global que ens serviran per comprovar la següent hipòtesis: - El petroli està correlacionat amb el l'IPC global. Per a fer-ho, hem utilitzat una simplificació degut a una manca de dades: hem simplificat l'IPC global fent la mitjana entre l'IPC d'Estats Units, Alemanya, Espanya, Rússia i Anglaterra.

```
[14]: """
def getCountryDataNoDF(country_code, initial_date, period, db,
    columns=None):
    ini_date = datetime.datetime.strptime(initial_date, "%d/%m/%Y")

    end_date = ini_date + datetime.timedelta(days=period)

    db.execute("SELECT * FROM dades_paisos WHERE "
```



```

        "id_pais=%s AND data_inici>=%s AND data_inici < %s",
        ↪(country_code, ini_date, end_date))

    result = db.fetchall()
    df = pd.DataFrame(result, columns=('id_pais', 'data_inici',
    ↪'data_final', 'PIB', 'IPC', 'deuta_publica'))

    return df

countries = ["US", "GB", "ES", "CA", "RU", "DE"]

dfs = []
for cc in countries:
    dfs.append(getCountryDataNoDF(cc, "01/01/2010", 99999, cursor))

alldata = pd.concat(dfs)
df4 = alldata[["data_inici", "IPC"]]
#["data_inici", "IPC"]]
df4.groupby("data_inici").mean()
"""

def getWorldIPC(initial_date, period, db, columns=None):
    ini_date = datetime.datetime.strptime(initial_date, "%d/%m/%Y")

    end_date = ini_date + datetime.timedelta(days=period)

    db.execute("SELECT data_inici,avg(IPC) FROM dades_paisos "
               "WHERE id_pais IN ('US', 'GB', 'ES', 'CA', 'RU', 'DE') "
               "GROUP BY data_inici "
               "HAVING data_inici>=%s AND data_inici < %s", (ini_date,
    ↪end_date))

    result = db.fetchall()
    df = pd.DataFrame(result, columns=('data_inici', 'ipc_avg'))
    df['data_inici'] = pd.to_datetime(df['data_inici']) # set dates to
    ↪datetime64
    df.set_index('data_inici', inplace=True)

```

```
return df["ipc_avg"]
```

```
[15]: cnx = mysql.connector.connect(user="root", password=password,
    database=database)
cursor = cnx.cursor()

world_IPC = getWorldIPC("01/01/2010", 9999, cursor)
oil = getPriceData('CL=F', "01-01-2016", 9999, cursor)['close']

cnx.commit()
cursor.close()

#group values by month
oil.index = pd.to_datetime(oil.index)
oil = oil.groupby(pd.Grouper(freq='M')).mean()
oil.index = oil.index.to_period('M').to_timestamp('D')

#Match index
oil = oil.drop(index=oil.index.difference(world_IPC.index))

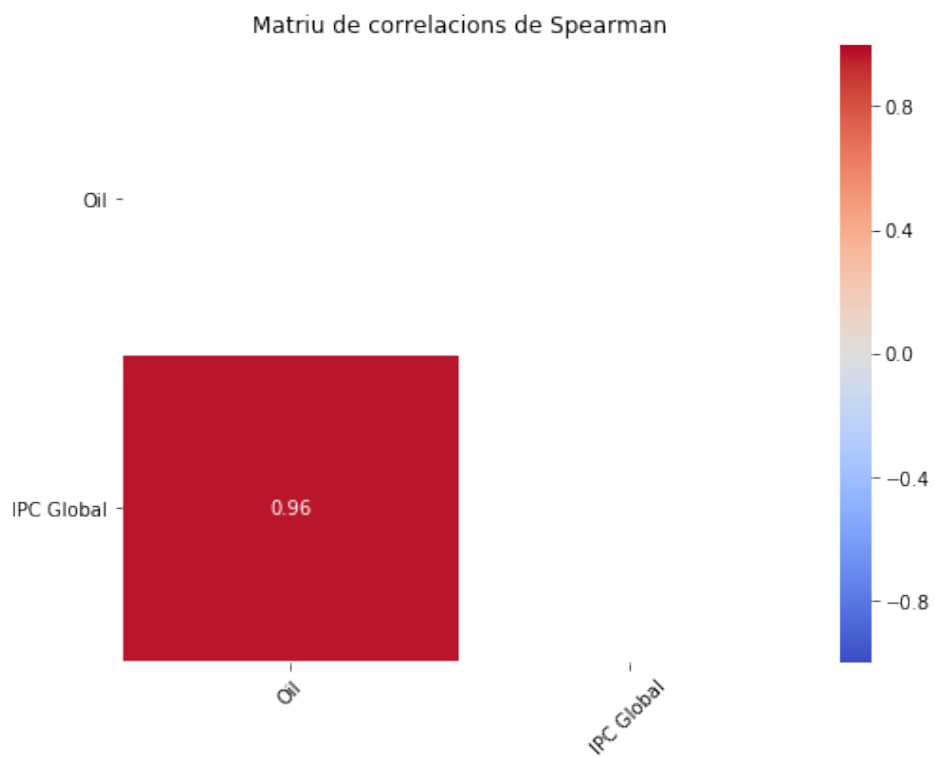
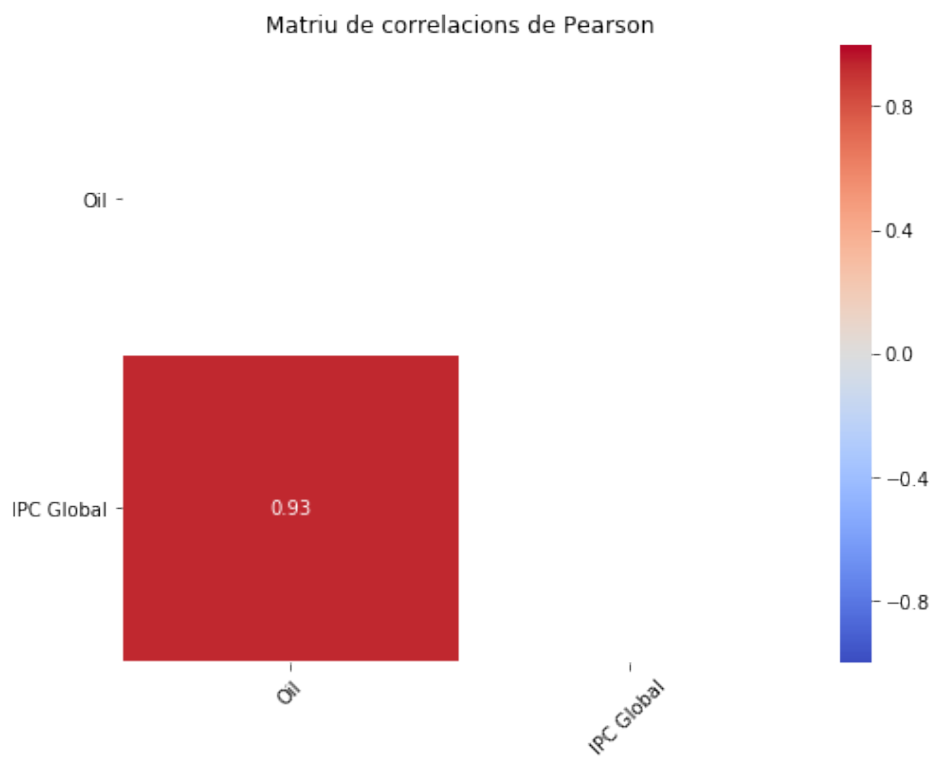
#print(oil.to_string())

#calculem les matrius

vector_corr_matrix = calculate_corr_matrix(["Oil", "IPC",
    Global"], oil, world_IPC)

#mostrem els resultats

plot_correlations(vector_corr_matrix[0], "Pearson")
plot_correlations(vector_corr_matrix[1], "Spearman")
```



El resultat és molt clar, L'IPC Global i el preu del petroli, estan estretament coorelacionats amb uns coeficients de corelació de 0.93 i 0.96 de pearson i spearman respectivament. Per tant, la hipòtesi ha estat verificada com a certa.

## 7.2 Tendències

Per esbrinar la veracitat d'hipotesis que impliquin tendències (per exemple, un cert actiu ha pujat a partir de cert event) utilitzarem la següent funció, que ens calcularà un parell d'indicadors comunment usats en anàlisis tècnic per calcular tendències i ens mostrarà el resultat en un gràfic, amb la possibilitat de marcar el dia de començament de dit event.

```
[16]: def trend_plot(data, d_day=None, title='', window_adx=14,
    ↪window_slow=26, window_fast=12, window_sign=9):
    adx = ta.trend.ADXIndicator(high=data["high"], low=data["low"],
    ↪close=data["close"], window=window_adx)
    macd = ta.trend.MACD(close=data["close"], window_slow=window_slow,
    ↪window_fast=window_fast, window_sign=window_sign)
    colors = ['g' if v >= 0 else 'r' for v in macd.macd_diff()]

    apds = [mpf.make_addplot(adx.adx_pos(), type='line', panel=1,
    ↪color='g', ylabel='ADX', y_on_right=True),
            mpf.make_addplot(adx.adx_neg(), type='line', panel=1,
    ↪color='r', secondary_y=False),
            mpf.make_addplot(adx.adx(), type='line', panel=1,
    ↪color='b', secondary_y=False),
            mpf.make_addplot(macd.macd_diff(), type='bar', panel=2,
    ↪color=colors, ylabel='MACD'),
            mpf.make_addplot(macd.macd(), type='line', panel=2,
    ↪color='b', secondary_y=False),
            mpf.make_addplot(macd.macd_signal(), type='line', panel=2,
    ↪color='orange', secondary_y=False)]

    fig, axlist = mpf.plot(data, type='candle', addplot=apds,
    ↪panel_ratios=(5,3), axtitle=title, returnfig=True)

    if d_day != None:
        for ax in axlist:
            ax.axvline(x=d_day, linestyle='--', color='purple')
```

```
mpf.show()
```

```
[17]: # IMPORTANT: execute to open database
cnx = mysql.connector.connect(user="root", password=password,
    ↪database=database)
cursor = cnx.cursor()
```

### 7.3 ADX

L'indicador ADX (Average Directional Index) serveix per mesurar com de pronunciada és una tendència i acostuma a representar-se juntament amb un indicador direccional positiu (+DI) i un indicador direccional negatiu (-DI). Quan l'indicador positiu està per sobre del negatiu considerem que hi ha una tendència alcista, i quan està per sota que hi ha una tendència baixista. Com més alt sigui el ADX més forta considerarem que és la tendència.

Els tres indicadors tenen un rang de valors entre 0 i 100. Utilitzem una finestra de 14 dies per calcular aquests tres indicadors.

Representarem el ADX en color blau, el +DI en color verd i el -DI en color vermell.

### 7.4 MACD

L'indicador MACD (Moving Average Convergence/Divergence) és la diferència entre una mitjana mòbil exponencial amb període de 12 dies i una mitjana mòbil exponencial amb període de 26 dies. S'acostuma a representar juntament amb una Línia de Senyal, que és una mitjana mòbil exponencial amb període de 9 dies, i un histograma, que es calcula fent la diferència entre el MACD i la Línia de Senyal. Aquests indicadors oscilen sobre l'eix del 0.

Quan l'indicador MACD passa per sobre de la Línia de Senyal (quan els valors de l'histograma són positius) es considera que hi ha una tendència alcista i quan passa per sota (quan l'histograma és negatiu) es considera que hi ha una tendència baixista. Com més alta sigui la diferència entre el MACD i la Línia de Senyal més pronunciada es considera que és la tendència (i els valors de l'historiograma són més extrems).

Representarem el MACD en color blau, la Línia de Senyal en color taronja i el historiograma en color verd (si es positiu) i vermell (si és negatiu).

### 7.5 Guerres i preus

#### 7.5.1 Guerra d'Ucraina i preu del petroli

En els dos mesos anteriors a l'inici de la guerra observem que la tendència del preu del petroli passa a ser a l'alça de manera força pronunciada, arribant a un pic poc després de la data

d'inici de la guerra i mantenint-se més o menys estable en els mesos següents, fins a 4 mesos després de l'inici de la guerra on comença una tendència de baixada de preu.

Així doncs, les dades semblen indicar que realment la guerra a tingut un impacte en el petroli, encarint el preu.

```
[18]: event = getEventData('ucraina', cursor)

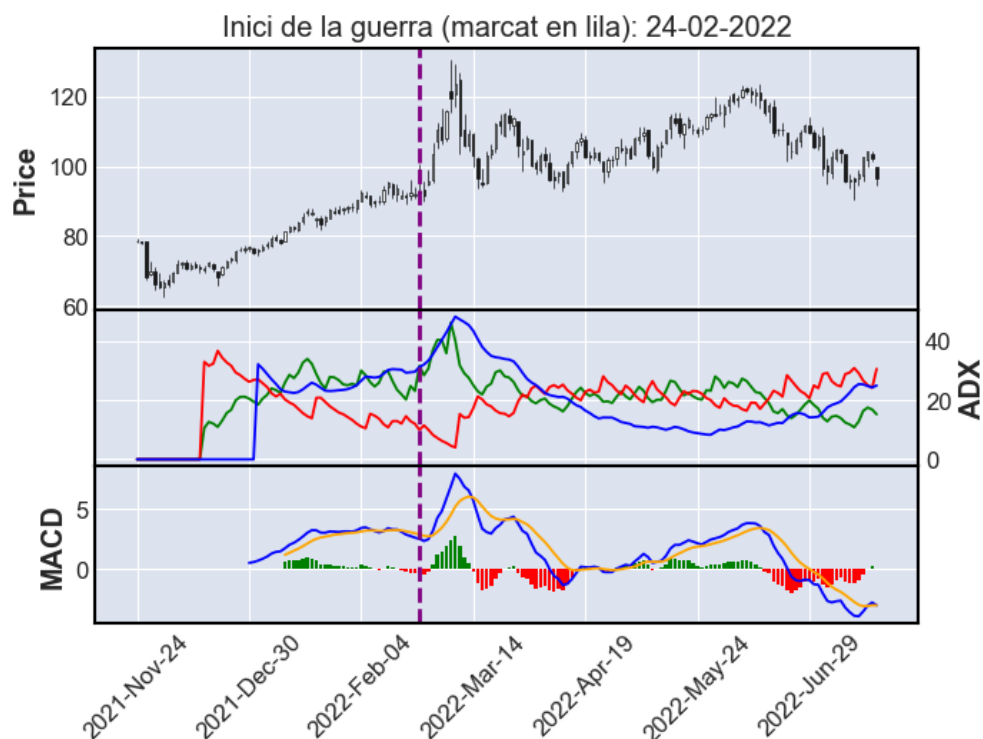
if len(event['data']) == 1:
    date_war = pd.to_datetime(event['data'].values[0])
    date_plot = date_war - pd.DateOffset(months=3)

    oil = getPriceData('CL=F', date_plot.strftime('%d-%m-%Y'), 240,
    cursor)

    title = "Inici de la guerra (marcat en lila): " + date_war.
    strftime('%d-%m-%Y')

    trend_plot(oil, d_day=oil.index.get_loc(date_war.
    strftime('%Y-%m-%d')), title=title)

else:
    print("No hem pogut recuperar el event de la base de dades")
```



### 7.5.2 Guerra de Síria i preu del petroli

Uns 20 dies abans de l'inici de la guerra hi ha un augment pronunciat amb tendència de pujada de preu, i aquesta tendència es manté fins un mes i mig més fins entrar en tendència de baixada de preus.

Així doncs, les dades semblen indicar que realment la guerra a tingut un impacte en el petroli, encarint el preu.

```
[19]: event = getEventData('syria', cursor)

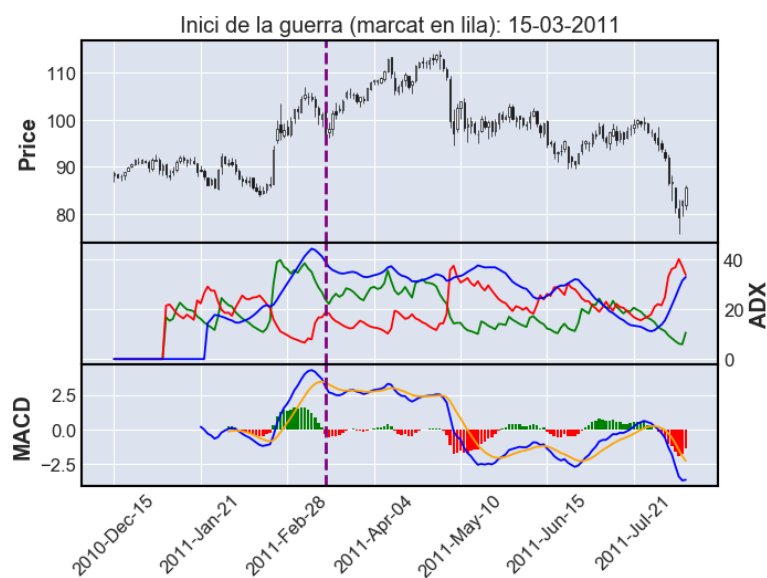
if len(event['data']) == 1:
    date_war = pd.to_datetime(event['data'].values[0])
    date_plot = date_war - pd.DateOffset(months=3)

    oil = getPriceData('CL=F', date_plot.strftime('%d-%m-%Y'), 240,
    cursor)

    title = "Inici de la guerra (marcat en lila): " + date_war.
    strftime('%d-%m-%Y')

    trend_plot(oil, d_day=oil.index.get_loc(date_war.
    strftime('%Y-%m-%d')), title=title)

else:
    print("No hem pogut recuperar el event de la base de dades")
```



### 7.5.3 Guerra d'Ucraina i inflació

No hem trobat prou dades per poder contrastar la hipòtesis amb seguretat. Tot i així, a partir de les dades que tenim sembla que la guerra de Ucraïna podria haver afectat al IPC global, donat que trobem un pic en el valor del mes següent a l'inici de la guerra.

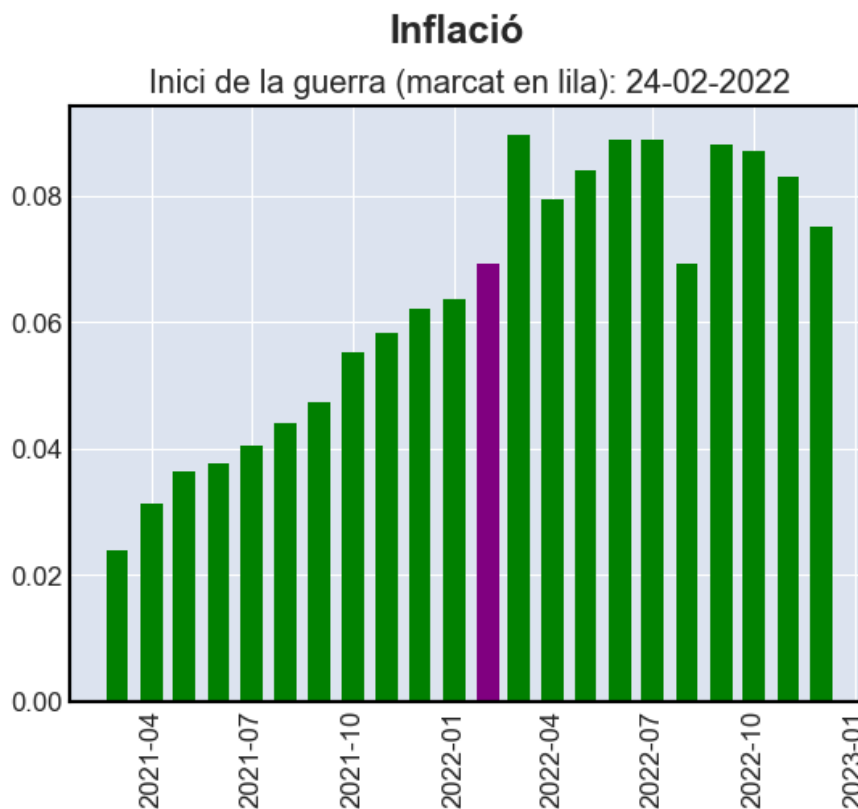
```
[20]: event = getEventData('ucraina', cursor)

if len(event['data']) == 1:
    date_war = pd.to_datetime(event['data'].values[0])
    date_plot = date_war - pd.DateOffset(months=12)

    inflacio = getWorldIPC(date_plot.strftime('%d/%m/%Y'), 9999, cursor)
    colors = ['purple' if d.month == date_war.month else 'green' for d_
    in inflacio.index]
    title = "Inici de la guerra (marcat en lila): " + date_war.
    strftime('%d-%m-%Y')
    plt.bar(inflacio.index, inflacio, color =colors, width = 20)
    plt.suptitle("Inflació", y=1)
    plt.title(title)
    plt.ticklabel_format(axis='y', style='plain')
    plt.xticks(rotation=90)
    plt.show()

else:
    print("No hem pogut recuperar el event de la base de dades")
```





#### 7.5.4 Guerra de Síria i inflació

No tenim dades de la inflació al 2011, de manera que no ho podem mesurar.

Guerra de Ucraïna i exportacions de Ucraïna

No hem pogut obtenir accés gratuït a les dades de les exportacions de Ucraïna en els últims dos anys, de manera que no hem pogut contrastar aquesta hipòtesi.

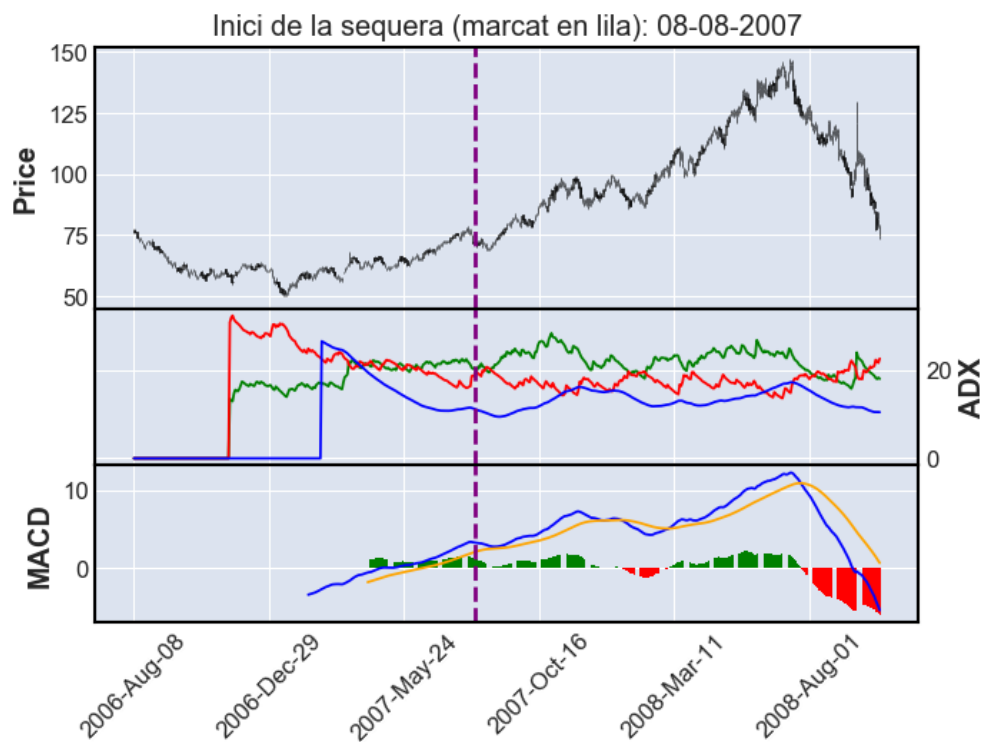
#### 7.6 Sequeres i suc de taronja

Podem observar en el gràfic el preu del suc de taronja al món i en una línia discontinua, la data on la sequera ja era molt severa. Es pot observar clarament que va tenir un impacte significatiu en el preu, ja que el preu es va duplicar en qüestió d'un any. Com que hi ha molts factors que arrelenteixen l'impacte de la sequera (com que les taronjes triguen un any en ser collides i que el suc de taronja es pot conservar durant molt de temps) hem decidit modificar les finestres temporals dels indicadors MACD i ADX per a que siguin mitjanes de 5 dies, per a tenir una millor imatge de la tendència.

```
[21]: event = getEventData('sequera', cursor)

if len(event['data']) == 1:
    date_drought = pd.to_datetime(event['data'].values[0])
    date_plot = date_drought - pd.DateOffset(months=12)

    orange_juice = getPriceData('CL=F', date_plot.strftime('%d-%m-%Y'),
    ↪800, cursor)
    title = "Inici de la sequera (marcat en lila): " + date_drought.
    ↪strftime('%d-%m-%Y')
    trend_plot(orange_juice, d_day=orange_juice.index.
    ↪get_loc(date_drought.strftime('%Y-%m-%d')), title=title,
    ↪window_adx=14*5, window_slow=26*5, window_fast=12*5, window_sign=9*5)
else:
    print("No hem pogut recuperar el event de la base de dades")
```



## 7.7 Aranzels de EEUU a ES i preu del pernil

Un altre cop, hem trobat dificultats a l'hora de trobar dades rellevants, ja que no es van trobar les dades d'exportacions de pernil, s'han usat les exportacions de conserves de carn i només hem obtingut dades interanuals per pocs anys. A més a més, també hi va haver el COVID poc després i per tant els resultats seran poc significatius. Tot i això, els resultats obtinguts, indiquen poca relació entre els aranzels i el preu de les conserves de carn.

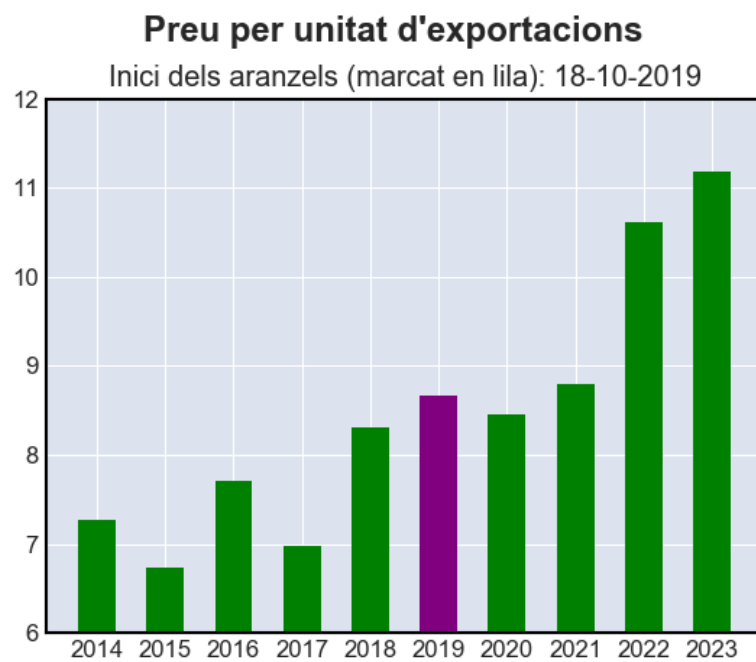
```
[22]: event = getEventData('aranzel', cursor)

if len(event['data']) == 1:
    date_aranzel = pd.to_datetime(event['data'].values[0])
    date_next_year = date_drought + pd.DateOffset(years=1)

    aranzels = getImportsData('ES', 'US', 'carne', cursor)
    colors = ['purple' if d.year == date_aranzel.year else 'green' for
    d in aranzels.index[0:10]]
    title = "Inici dels aranzels (marcat en lila): " + date_aranzel.
    strftime('%d-%m-%Y')
    plt.bar(aranzels.index[0:10], aranzels['quantitat'][0:10], color
    =colors, width = 200)
    plt.suptitle("Quantitat d'exportacions", y=1)
    plt.title(title)
    plt.ticklabel_format(axis='y', style='plain')
    plt.show()

    price = aranzels['preu'][0:10] / aranzels['quantitat'][0:10]
    plt.bar(aranzels.index[0:10], price, color =colors, width = 200)
    plt.suptitle("Preu per unitat d'exportacions", y=1)
    plt.title(title)
    plt.ticklabel_format(axis='y', style='plain')
    plt.ylim([6,12])
    plt.show()

else:
    print("No hem pogut recuperar el event de la base de dades")
```



```
[23]: # IMPORTANT: execute to close database
cnx.commit()
cursor.close()
```

## 8 Instal·lació de la base de dades al núvol

Ens vam plantejar, com a tasca opcional pel projecte, pujar al núvol la base de dades, per poder experimentar com seria una base de dades en remot. Vam, doncs, configurar una instància (gratuïta per ser estudiant) de Azure MySQL a la plataforma Azure. D'aquesta manera, tot i que no li hem acabat de donar un ús molt rellevant, vam experimentar el que seria connectar-se amb una base de dades en remot mitjançant el workbench i amb el client de terminal.

Hem pogut comprovar també com un servei d'aquestes característiques té un munt d'opcions de configuració i sobre tot de seguretat (fet que ens va, també, dificultar la importació de dades a través de fitxers al núvol). Vam importar les dades simplement amb un backup de la base de dades plena en local.

## 9 Conclusió final

Els objectius plantejats a l'inici del treball han estat posats a prova i aquests són els resultats resumits:

1. No hi ha correlació entre la fusta i el petroli.
2. Existeix una correlació significativa entre el gas natural i la soja.
3. La guerra d'Ucraïna ha provocat un encariment del preu del petroli.
4. No s'han obtingut prous dades per a concloure sobre l'impacte de la guerra d'Ucraïna en les exportacions.
5. No s'han pogut obtenir conclusions sobre la relació entre la guerra de Síria i la inflació.
6. Les sequeres tenen un impacte significatiu en l'encariment del suc de taronja.
7. Hi ha una correlació estreta entre el preu del petroli i l'IPC global.
8. Hi ha una correlació estreta entre el PIB Anglès i el PIB Europeu.
9. No s'han pogut establir conclusions sobre la relació entre els aranzels de EEUU a Espanya i el preu del pernil.

Dit això, l'objectiu principal del treball i de l'assignatura, era treballar amb bases de dades i aquest objectiu s'ha assolit amb èxit. S'ha demostrat una comprensió i aplicació adequades de SQL, i s'ha aconseguit posar en funcionament la base de dades tant en un entorn local com en un servidor d'Azure. Això ha demostrat habilitats tècniques i coneixements en la gestió de bases de dades i en l'ús de tecnologies de núvol.

A més, hem tingut també la oportunitat de treballar amb llibreries de python de manipulació i anàlisi de dades, com són pandas o matplotlib, per realitzar un anàlisi estadístic de les dades. En particular, hem après com treballar amb sèries temporals i com analitzar-ne les tendències i correlacions, fent ús d'estadístics normalment més habituals en l'àmbit financer.

D'altra banda, ens hem trobat i hem hagut d'anar sorteiant els que pensem que series problemes habituals en la realització d'un projecte com aquest: cerca de les dades, formatar-les correctament, configuració del gestor SQL i de les diferents taules, recuperació de les dades perquè encaixin en el nostre llenguatge de programació (i.e. python), entre d'altres.