

Creación de un dashboard para usuarios del ticket digital de Mercadona con visualización gráfica de datos: evolución de precios por producto, gastos por categoría de alimentación y ventanas temporales de gastos

Santiago Sánchez Sans

IES Abastos

6 junio 2025

Contenido

1 Introducción

2 Diseño

- Requisitos
 - Diagramas de sistemas

3 Desarrollo

- Entornos de desarrollo
 - Despliegue
 - Spring Boot: gestión usuarios
 - FastAPI: parseo de tickets
 - FastAPI: exposición tickets (+ dashboard)
 - Front-end: Vanilla JS

4 Conclusiones

Contenidos

1 Introducción

2 Diseño

- Requisitos
- Diagramas de sistemas

3 Desarrollo

- Entornos de desarrollo
- Despliegue
- Spring Boot: gestión usuarios
- FastAPI: parseo de tickets
- FastAPI: exposición tickets (+ dashboard)
- Front-end: Vanilla JS

4 Conclusiones

1. Introducción

■ Identificación de necesidades:

- Usuario del ticket digital → no tiene informes de sus datos.

■ Objetivos:

- Proporcionar al usuario del ticket digital una herramienta que muestre en gráficos visuales:

- **Evolución de precios** (inflación) a lo largo del tiempo en los productos habitualmente obtenidos en el mismo establecimiento¹.
 - **Evolución del gasto** total del usuario a lo largo del tiempo por períodos temporales.

¹ La evolución de precios se mostrará solamente para un mismo centro de Mercadona, dado que distintos centros pueden cambiar los nombres de los productos (por ejemplo, en Cataluña...)

Contenidos

1 Introducción

2 Diseño

- Requisitos
- Diagramas de sistemas

3 Desarrollo

- Entornos de desarrollo
- Despliegue
- Spring Boot: gestión usuarios
- FastAPI: parseo de tickets
- FastAPI: exposición tickets (+ dashboard)
- Front-end: Vanilla JS

4 Conclusiones

Requisitos

Requisitos de los usuarios

Que los usuarios tengan una cuenta de gmail con tickets digitales de Mercadona dentro e, idealmente, tenga decenas de tickets digitales: idealmente con compras estables y productos de adquisición recurrentes. El requisito indispensable es tener un mínimo de dos tickets digitales distintos.

Requisitos

Requisitos funcionales

REQUISITO A: Mostrar **evolución de los precios** de los productos unitarios adquiridos con más frecuencia (visualizable en un gráfico donde en X tendremos el tiempo y en Y el precio en euros).

REQUISITO B: Mostrar gasto total en distintas ventanas temporales del usuario: períodos de 1, 3, 6 meses y un año; independientemente del centro de Mercadona en el que se compre (todos juntos).

REQUISITO C: Al lado del gasto total anterior se incluirá un **diagrama de sectores** desglosando porcentaje de dinero gastado en 13 categorías ([click para ver categorías](#))

Requisitos

Requisitos funcionales (cont.)

REQUISITO D²: Los PDFs descargados del correo del usuario se almacenarán en una carpeta local del ordenador del usuario.

REQUISITO E³: El sistema front-end y back-end de registro permitirá redirigir a los usuarios rápidamente a un registro de forma inteligente. Nos inspiraremos en el sistema de registro e iniciar sesión de NetFlix. Ver diagrama enrutamiento.

²Requisito añadido después de la presentación inicial del proyecto.

³Requisito añadido después de la presentación del proyecto.

Requisitos

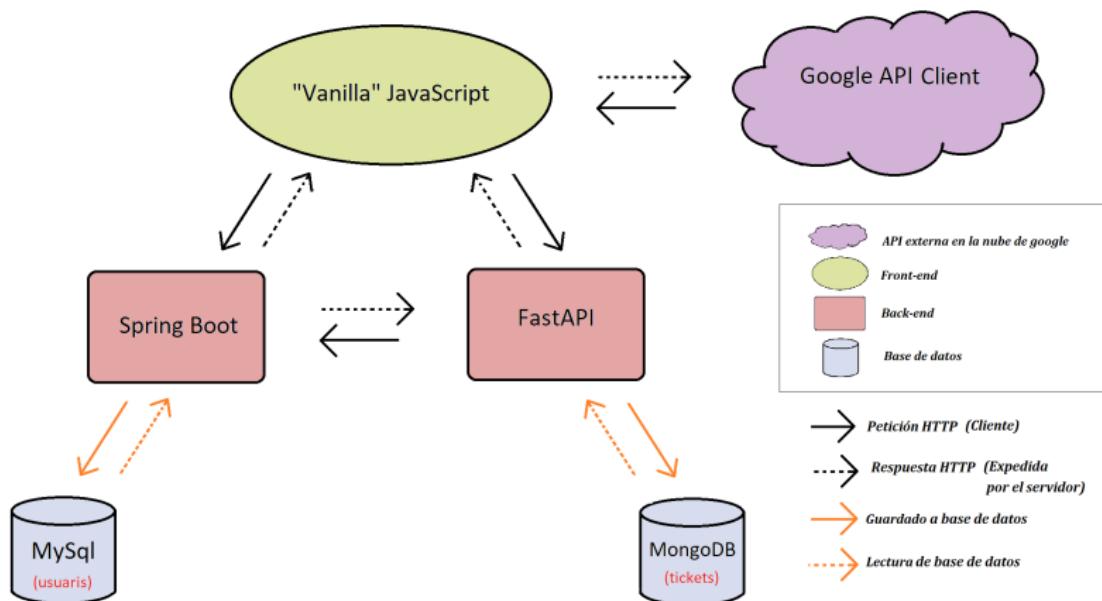
Requisitos funcionales (RESUMEN)

De los requisitos al diseño (anticipo de lo que será el dashboard):

- evolución de precios por producto → “**inflalyzer**”
- gastos por categoría de alimentación → “**categoryzer**”
- ventanas temporales de gastos → “**intervalizer**”

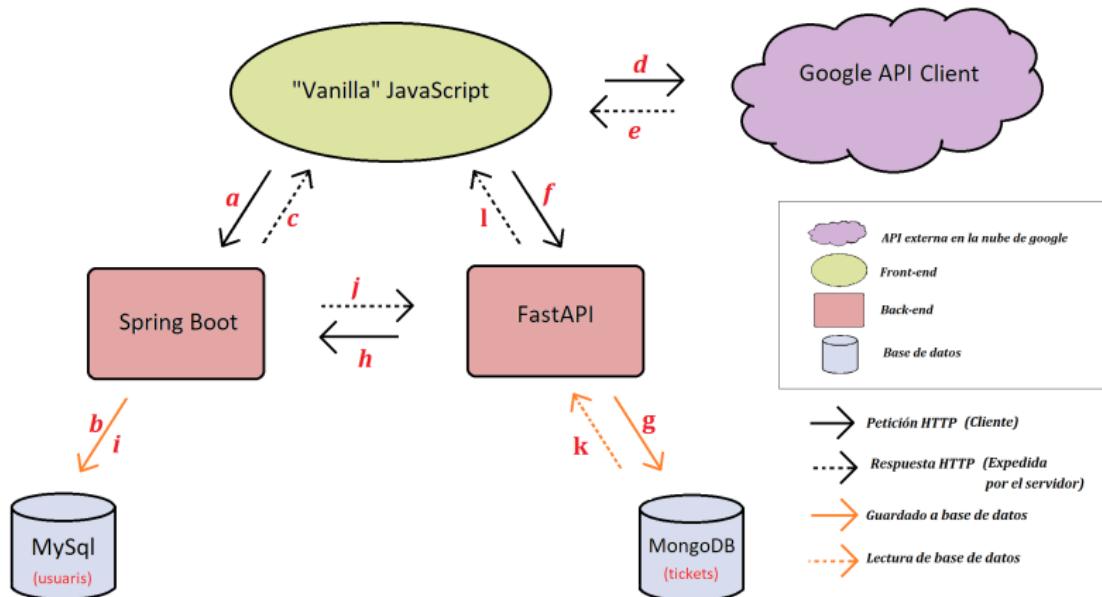
Diagramas de sistemas

Diagrama general



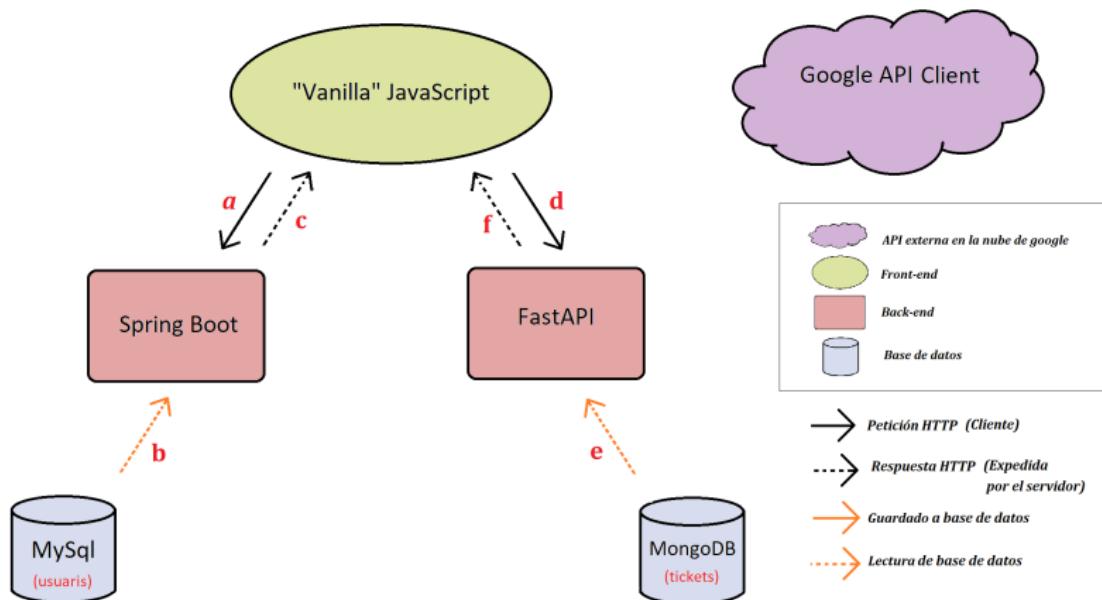
Diagramas de sistemas

registro



Diagramas de sistemas

inicio de sesión



Contenidos

1 Introducción

2 Diseño

- Requisitos
- Diagramas de sistemas

3 Desarrollo

- Entornos de desarrollo
- Despliegue
- Spring Boot: gestión usuarios
- FastAPI: parseo de tickets
- FastAPI: exposición tickets (+ dashboard)
- Front-end: Vanilla JS

4 Conclusiones

Entornos de desarrollo

Entornos de desarrollo

Editor / Herramienta	Puerto⁴⁾
IntelliJ IDEA (Java, SpringBoot)	8080
VSCode (HTML, CSS, JS con Live Server)	5500
VSCode (Python, con FastAPI ⁵⁾	8000
MySQL Workbench	3306
MongoDB Compass	27017

Table: Entornos de desarrollo y puertos utilizados para despliegue en local

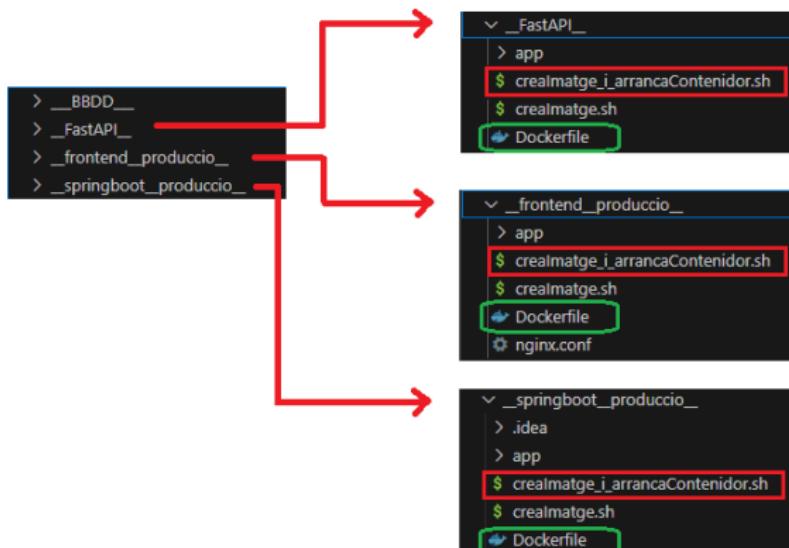
⁴ El host es localhost

⁵No depende del editor de código

Despliegue

Despliegue

Se ha automatizado la creación de imágenes e instanciado de contenedores para cada microservicio. PUERTOS: ¡idem!



Despliegue

Despliegue (cont.)

Imagen original	puerto
openjdk:17-alpine	8080
nginx:alpine	5500 ⁶
Python:alpine (DF)	8000

Table: Imágenes docker base y puertos donde instanciamos su contenedor

base de datos	puerto
MySQL	3306
MongoDB	27017

Table: Bases de datos: no contenerizadas

⁶localhost no sirve; usar 127.0.0.1 en navegador para ver index.html

Despliegue

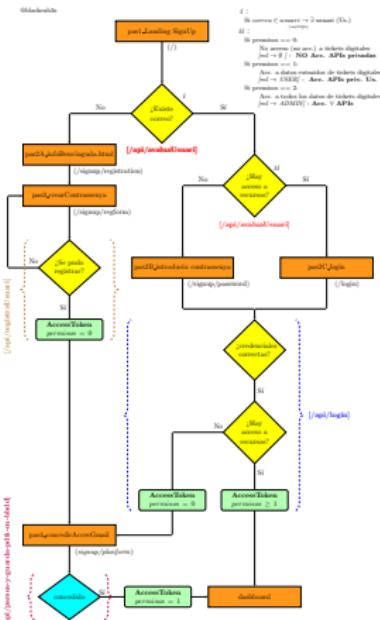
CONTINUAR PER 3.4 DE LA MEMORIA EN APARTAT DESARROLLO

ometre dockeritzacio que surti a desarollo de la memoria perque ja s'ha posat lo basic a disseny per no repetir. Posar sobretot estructures projectes i NO oblidar el diagrama d'enrutament.

Spring Boot: gestión usuarios



Spring Boot: gestión usuarios



Spring Boot: gestión usuarios



Expedición de un token de acceso (permisos = 1)

Entrada de un idUsuario proporcionado por una solicitud post de FastAPI (permisos=0)

```
//PRE: existeix un idUsuari en la base de dades amb permisos a 0.  
//POST: canviem permisos idUsuari a 1 en bbdd + retorno JWT fresc (amb permisos 1)  
public String obtenerTokenPermisosPerDashboard(int idUsuari) {  
    Optional<Usuari> usuariOptional = repoUsuari.findById(idUsuari);  
    Usuari u = usuariOptional.get();  
    u.setPermisos((byte) 1);  
    repoUsuari.save(u);  
    return this.generaTokenAccesoPerUsuariParticular(u.getCorreuElectronico());  
}
```

Gracias a JPA y su ORM.

Figure: Función de servicio en Spring Boot para expedir un acceso con el id de usuario proporcionado por FastAPI (solicitud cliente cuando FastAPI ya tiene tickets persistidos). Nótese que en 3 líneas de código tenemos la persistencia en mySQL hecha. ¿Función save?.

Spring Boot: gestión usuarios



```
@Repository  
public interface UsuariRepositori extends  
JpaRepository<Usuari, Integer> {
```

Figure: Con JpaRepository tenemos funciones para persistir en base de datos sin tener que definir consultas (la función save de la diapositiva anterior). Al hacer nuestra interface para operaciones de persistencia solamente debememos extender de JpaRepository e indicar la clase con la que hacemos el ORM y su clave primaria

FastAPI: parseo de tickets





FastAPI: parseo de tickets

LEYENDA



Delimitadores

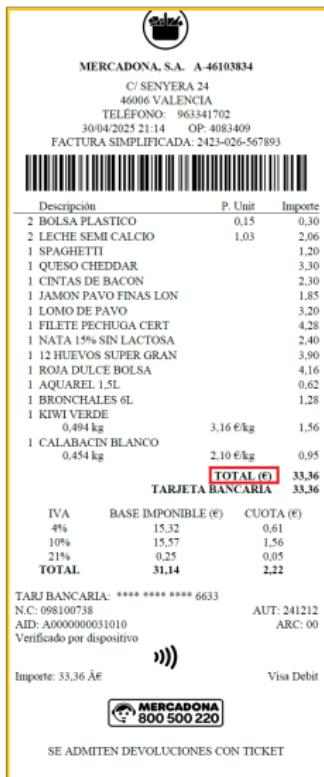


ROIs

MERCADONA, S.A. A-46103834			
C/ SENYERA 24			
46006 VALENCIA			
TELEFONO: 963341702			
30/04/2025 21:14 OP: 4083409			
FACTURA SIMPLIFICADA: 2423-026-567893			
			
Descripción	P. Unit	Importe	
2 BOLSA PLASTICO	0,15	0,30	
2 LECHE SEMI CALCIO	1,03	2,06	
1 SPAGHETTI		1,20	
1 QUESO CHEDDAR		3,30	
1 CINTAS DE BACON		2,30	
1 JAMON PAVO FINAS LON		1,85	
1 LOMO DE PAVO		3,20	
1 FILETE PECHUGA CERT		4,28	
1 NATA 15% SIN LACTOSA		2,40	
12 HUEVOS SUPER GRAN		3,90	
1 ROJA DULCE BOLSA		4,16	
1 AQUAREL 1,5L		0,62	
1 BRONCHALES 6L		1,28	
1 KIWI VERDE			
0,494 kg	3,16 €/kg	1,56	
1 CALABACIN BLANCO			
0,454 kg	2,10 €/kg	0,95	
	TOTAL (€)	33,36	
	TARJETA BANCARIA	33,36	
IVA	BASE IMPONIBLE (€)	CUOTA (€)	
4%	15,32	0,61	
10%	15,57	1,56	
21%	0,25	0,05	
TOTAL	31,14	2,22	
TARIJ BANCARIA: ***** 6633			
N.C: 098100738		AUT: 241212	
AID: A0000000031010		ARC: 00	
Verificado por dispositivo			
			
Importe: 33,36 €		Visa Debit	
			
SE ADMITEN DEVOLUCIONES CON TICKET			

FastAPI: parseo de tickets

LEYENDA

 Delimitador


```
def pdf_to_text(file_path):
    try:
        text = ""
        with open(file_path, "rb") as file:
            reader = PyPDF2.PdfReader(file)
            for page_num in range(len(reader.pages)):
                page = reader.pages[page_num]
                text += page.extract_text() + "\n"
    return text
except PyPDF2.errors.PdfReadError:
    return "ErrorPdfText"
```

`textPDF = pdf_to_text(doc)`

`importeTotal = float(textPDF['importeTotal'])`

FastAPI: parseo de tickets

LEYENDA



Delimitador

MERCADONA, S.A. A-46103834

C/ SENYERA 24
46006 VALENCIA
TELÉFONO: 963341702
30/04/2025 21:14 - OP: 4083409
FACTURA SIMPLIFICADA: 2423-026-567893

Descripción	P. Unit	Importe
2 BOLSAS PLASTICO	0,15	0,30
2 LECHE SEMI CALCIO	1,03	2,06
1 SPAGHETTI		1,20
1 QUESO CHEDDAR		3,30
1 CINTAS DE BACON		2,30
1 JAMON PAVO FINAS LON		1,85
1 LOMO DE PAVO		3,20
1 FILETE PECHUGA CERT		4,28
1 NATA 15% SIN LACTOSA		2,40
12 HUEVOS SUPER GRAN		3,90
1 ROJA DULCE BOLSA		4,16
1 AQUAREL 1,5L		0,62
1 BRONCHALES 6L		1,28
1 KIWI VERDE 0,494 kg	3,16 €/kg	1,56
1 CALABACIN BLANCO 0,454 kg	2,10 €/kg	0,95
TOTAL (€)	33,36	
TARJETA BANCARIA	33,36	

IVA	BASE IMPONIBLE (€)	CUOTA (€)
4%	15,32	0,61
10%	15,57	1,56
21%	0,25	0,05
TOTAL	31,14	2,22

TARJ BANCARIA: **** * 6633
NC: 098100738
AID: A0000000031010
Verificado por dispositivo

Importe: 33,36 €

SE ADMITEN DEVOLUCIONES CON TICKET

MERCADONA
800 500 220

separacionTicketPorTotal(0)

separacionTicketPorTotal(1)



FastAPI: parseo de tickets

LEYENDA

 Delimitador



MERCADONA, S.A. A-46103834

C/ SENYERA 24
46006 VALENCIA
TELÉFONO: 963341702
30/04/2025 21:14 - OP: 4083409
FACTURA SIMPLIFICADA: 2423-026-567893



Descripción	P. Unit	Importe
2 BOLSA PLASTICO	0,15	0,30
2 LECHE SEMI CALCIO	1,03	2,06
1 SPAGHETTI		1,20
1 QUESO CHEDDAR		3,30
1 CINTAS DE BACON		2,30
1 JAMON PAVO FINAS LON		1,85
1 LOMO DE PAVO		3,20
1 FILETE PECHUGA CERT		4,28
1 NATA 15% SIN LACTOSA		2,40
1 12 HUEVOS SUPER GRAN		3,90
1 ROJA DULCE BOLSA		4,16
1 AQUAREL 1,5L		0,62
1 BRONCHALES 6L		1,28
1 KIWI VERDE 0,494 kg	3,16 €/kg	1,56
1 CALABACIN BLANCO 0,454 kg	2,10 €/kg	0,95
TOTAL (€)	33,36	
TARJETA BANCARIA		33,36
IVA	BASE IMPONIBLE (€)	CUOTA (€)
4%	15,32	0,61
10%	15,57	1,56
21%	0,25	0,05
TOTAL	31,14	2,22

TARJ BANCARIA: **** * 6633
N.C: 098100738
AID: A0000000031010
Verificado por dispositivo 

AUT: 241212
ARC: 00


Importe: 33,36 €


SE ADMITEN DEVOLUCIONES CON TICKET





FastAPI: parseo de tickets

LEYENDA

- Delimitador
- ROI



MERCADONA, S.A. A-46103834

C/ SENYERA 24
46006 VALENCIA
TELÉFONO: 963341702
30/04/2024 21:14 - OP: 4083409
FACTURA SIMPLIFICADA: 2423-026-567893



Descripción	P. Unit	Importe
2 BOLSA PLASTICO	0,15	0,30
2 LECHE SEMI CALCIO	1,03	2,06
1 SPAGHETTI		1,20
1 QUESO CHEDDAR		3,30
1 CINTAS DE BACON		2,30
1 JAMON PAVO FINAS LON		1,85
1 LOMO DE PAVO		3,20
1 FILETE PECHUGA CERT		4,28
1 NATA 15% SIN LACTOSA		2,40
1 12 HUEVOS SUPER GRAN		3,90
1 ROJA DULCE BOLSA		4,16
1 AQUAREL 1,5L		0,62
1 BRONCHALES 6L		1,28
1 KIWI VERDE 0,494 kg	3,16 €/kg	1,56
1 CALABACIN BLANCO 0,454 kg	2,10 €/kg	0,95
TOTAL (€)	33,36	
TARJETA BANCARIA	33,36	

IVA	BASE IMPONIBLE (€)	CUOTA (€)
4%	15,32	0,61
10%	15,57	1,56
21%	0,25	0,05
TOTAL	31,14	2,22

TARJ BANCARIA: ***** ***** 6633
N.C: 098100738
AID: A0000000031010
Verificado por dispositivo 

Importe: 33,36 € 

Vista Debit


MERCADONA
800 500 220

SE ADMITEN DEVOLUCIONES CON TICKET

```

separacionTicketPerTotal = tarjetaTicketPerTotal[0]
separacionTicketPerTotal[0] = separacionTicketPerTotal[0].replace(" ", "")
separacionTicketPerTotal[1] = separacionTicketPerTotal[1].replace(" ", "")

separacionTicketPerTotal[0] + separacionTicketPerTotal[1]

```



FastAPI: parseo de tickets

LEYENDA



Delimitador



MERCADONA, S.A. A-46103834

C/ SENYERA 24
46006 VALENCIA
TELÉFONO: 963341702
30/04/2025 21:14 - OP: 4083409
FACTURA SIMPLIFICADA: 2423-026-567893



Descripción	P. Unit	Importe
2 BOLSAS PLASTICO	0,15	0,30
2 LECHE SEMI CALCIO	1,03	2,06
1 SPAGHETTI		1,20
1 QUESO CHEDDAR		3,30
1 CINTAS DE BACON		2,30
1 JAMON PAVO FINAS LON		1,85
1 LOMO DE PAVO		3,20
1 FILETE PECHUGA CERT		4,28
1 NATA 15% SIN LACTOSA		2,40
12 HUEVOS SUPER GRAN		3,90
1 ROJA DULCE BOLSA		4,16
1 AQUAREL 1,5L		0,62
1 BRONCHALES 6L		1,28
1 KIWI VERDE		
0,494 kg	3,16 €/kg	1,56
1 CALABACIN BLANCO	2,10 €/kg	0,95
TOTAL (€)	33,36	
TARJETA BANCARIA		33,36

IVA	BASE IMPONIBLE (€)	CUOTA (€)
4%	15,32	0,61
10%	15,57	1,56
21%	0,25	0,05
TOTAL	31,14	2,22

TARJ BANCARIA: **** * 6633
NC: 098100738
AID: A0000000031010
Verificado por dispositivo

Importe: 33,36 €

Visa Debit


MERCADONA
800 500 220

SE ADMITEN DEVOLUCIONES CON TICKET

`separacionTicketPorTotal = separacionTicketPorTotal[0]``separacionTicketPorTotal[0]``separacionTicketPorTotal[1]`

String de una sola línea con la parte verde del ticket digital dentro

FastAPI: parseo de tickets



MERCADONA, S.A. A-46103834

C/ SENYERA 24

46006 VALENCIA

TELÉFONO: 963341702

30/04/2025 21:14 OP: 4083409

FACTURA SIMPLIFICADA: 2423-026-567893



Descripción	P. Unit	Importe
2 BOLSA PLASTICO	0,15	0,30
2 LECHE SEMI CALCIO	1,03	2,06
1 SPAGHETTI		1,20
1 QUESO CHEDDAR		3,30
1 CINTAS DE BACON		2,30
1 JAMON PAVO FINAS LON		1,85
1 LOMO DE PAVO		3,20
1 FILETE PECHUGA CERT		4,28
1 NATA 15% SIN LACTOSA		2,40
1 12 HUEVOS SUPER GRAN		3,90
1 ROJA DULCE BOLSA		4,16
1 AQUAREL 1,5L		0,62
1 BRONCHALES 6L		1,28
1 KIWI VERDE 0,494 kg	3,16 €/kg	1,56
1 CALABACIN BLANCO 0,454 kg	2,10 €/kg	0,95

```
ll_liniesTicket = separacioTicketPerTOTAL[0].split("\n")
for i in range(len(ll_liniesTicket)):
    print(ll_liniesTicket[i])
```

MERCADONA, S.A. A-46108384
C/ SENYERA 24
46006 VALENCIA
TELÉFONO: 963341702
30/04/2025 21:14 OP: 4083409
FACTURA SIMPLIFICADA: 2423-026-567893
Descripción P. Unit Importe
2BOLSA PLASTICO 0,15 0,30
2LECHE SEMI CALCIO 1,03 2,06
1SPAGHETTI 1,20
1QUESO CHEDDAR 3,30
1CINTAS DE BACON 2,30
1JAMON PAVO FINAS LON 1,85
1LOMO DE PAVO 3,20
1FILETE PECHUGA CERT 4,28
1NATA 15% SIN LACTOSA 2,40
112 HUEVOS SUPER GRAN 3,90
1ROJA DULCE BOLSA 4,16
1FILETE PECHUGA CERT 4,28
1NATA 15% SIN LACTOSA 2,40
112 HUEVOS SUPER GRAN 3,90
1ROJA DULCE BOLSA 4,16
1AQUAREL 1,5L 0,62
1BRONCHALES 6L 1,28
1KIMI VERDE
0,494 kg 3,16 €/kg 1,56
1CALABACIN BLANCO
0,454 kg 2,18 €/kg 0,95

separacioTicketPerTOTAL[0]

FastAPI: parseo de tickets



MERCADONA, S.A. A-46103834

C/ SENYERA 24
46006 VALENCIA

TELEFONO: 963341702

30/04/2025 21:14 OP: 4083409

FACTURA SIMPLIFICADA: 2423-026-567893



Descripción	P. Unit	Importe
2 BOLSA PLASTICO	0,15	0,30
2 LECHE SEMI CALCIO	1,03	2,06
1 SPAGHETTI		1,20
1 QUESO CHEDDAR		3,30
1 CINTAS DE BACON		2,30
1 JAMON PAVO FINAS LON		1,85
1 LOMO DE PAVO		3,20
1 FILETE PECHUGA CERT		4,28
1 NATA 15% SIN LACTOSA		2,40
1 12 HUEVOS SUPER GRAN		3,90
1 ROJA DULCE BOLSA		4,16
1 AQUAREL 1,5L		0,62
1 BRONCHALES 6L		1,28
1 KIWI VERDE 0,494 kg	3,16 €/kg	1,56
1 CALABACIN BLANCO 0,454 kg	2,10 €/kg	0,95

```
ll_liniesTicket = separacioTicketPerLinea
for i in range(len(ll_liniesTicket)):
    print(ll_liniesTicket[i])
```

MERCADONA, S.A. A-46103834
C/ SENYERA 24
46006 VALENCIA

[...]

```
carrer, CP_ciutat = ll_liniesTicket[1], ll_liniesTicket[2]
direccioSuper = carrer + " " + CP_ciutat
```

separacioTicketPerTOTAL[9]

FastAPI: parseo de tickets



MERCADONA, S.A. A-46103834

C/ SENYERA 24

46006 VALENCIA

TELÉFONO: 963341702

30/04/2025 21:14 OP: 4083409

FACTURA SIMPLIFICADA: 2423-026-567893



Descripción	P. Unit	Importe
2 BOLSA PLASTICO	0,15	0,30
2 LECHE SEMI CALCIO	1,03	2,06
1 SPAGHETTI		1,20
1 QUESO CHEDDAR		3,30
1 CINTAS DE BACON		2,30
1 JAMON PAVO FINAS LON		1,85
1 LOMO DE PAVO		3,20
1 FILETE PECHUGA CERT		4,28
1 NATA 15% SIN LACTOSA		2,40
1 12 HUEVOS SUPER GRAN		3,90
1 ROJA DULCE BOLSA		4,16
1 AQUAREL 1,5L		0,62
1 BRONCHALES 6L		1,28
1 KIWI VERDE 0,494 kg	3,16 €/kg	1,56
1 CALABACIN BLANCO 0,454 kg	2,10 €/kg	0,95

```
ll_liniesTicket = separacioTicketPerTOTAL[0].split("\n")
for i in range(len(ll_liniesTicket)):
    print(ll_liniesTicket[i])
```

MERCADONA, S.A. A-46103834
C/ SENYERA 24
46006 VALENCIA
TELÉFONO: 963341702
30/04/2025 21:14 OP: 4083409
FACTURA SIMPLIFICADA: 2423-026-5678
Descripción P. Unid. Terciente

```
dataOperacio = ll_liniesTicket[4]
liniaFactura = ll_liniesTicket[5]
```

separacionTicketPerTOTAL [8]

FastAPI: parseo de tickets



MERCADONA, S.A. A-46103834

C/ SENYERA 24

46006 VALENCIA

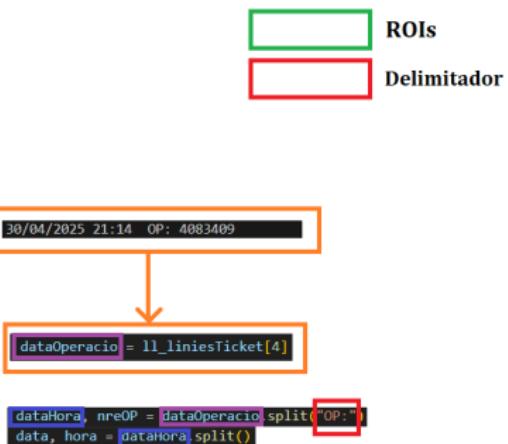
TELÉFONO: 963341702

30/04/2025 21:14

FACTURA SIMPLIFICADA: 2423-026-567893



Descripción	P. Unit	Importe
2 BOLSA PLASTICO	0,15	0,30
2 LECHE SEMI CALCIO	1,03	2,06
1 SPAGHETTI		1,20
1 QUESO CHEDDAR		3,30
1 CINTAS DE BACON		2,30
1 JAMON PAVO FINAS LON		1,85
1 LOMO DE PAVO		3,20
1 FILETE PECHUGA CERT		4,28
1 NATA 15% SIN LACTOSA		2,40
1 12 HUEVOS SUPER GRAN		3,90
1 ROJA DULCE BOLSA		4,16
1 AQUAREL 1,5L		0,62
1 BRONCHALES 6L		1,28
1 KIWI VERDE 0,494 kg	3,16 €/kg	1,56
1 CALABACIN BLANCO 0,454 kg	2,10 €/kg	0,95



```
data, hora = datahora.split()
```

```
dataHora, nreOP = dataOperac.  
data, hora = dataHora.split('
```

```
dataOperacio = ll_liniesTicket[4]
```

30/04/2025 21:14 OP: 4083409

14

30/04/2025

FastAPI: parseo de tickets



MERCADONA S.A. A-46103834

C/ SENYERA 24

46006 VALENCIA

TELÉFONO: 963341702

30/04/2025 21:14 OP: 4083409

FACTURA SIMPLIFICADA: 3423-026-567803



Descripción	P. Unit	Importe
2 BOLSA PLASTICO	0,15	0,30
2 LECHE SEMI CALCIO	1,03	2,06
1 SPAGHETTI		1,20
1 QUESO CHEDDAR		3,30
1 CINTAS DE BACON		2,30
1 JAMON PAVO FINAS LON		1,85
1 LOMO DE PAVO		3,20
1 FILETE PECHUGA CERT		4,28
1 NATA 15% SIN LACTOSA		2,40
1 12 HUEVOS SUPER GRAN		3,90
1 ROJA DULCE BOLSA		4,16
1 AQUAREL 1,5L		0,62
1 BRONCHALES 6L		1,28
1 KIWI VERDE 0,494 kg	3,16 €/kg	1,56
1 CALABACIN BLANCO 0,454 kg	2,10 €/kg	0,95

30/04/2025 21:14 OP: 4083409
FACTURA SIMPLIFICADA: 2423-026-567893

```
dataOperacio = ll_liniesTicket[4]
liniaFactura = ll_liniesTicket[5]
```

```
dataHora = dataOperacion.split('OP:')
```

```
# CREEM UNA CLAU PRIMÀRIA PER TICKET QUE NO S'Hauria de REPETIR MAI  
nreFactura = liniaFactura.split("FACTURA SIMPLIFICADA:")[1].strip()  
nreOP = nreOP.strip()  
clauPrimaria = nreFactura + " OP" + nreOP
```

2423-026-567893

4083409

2423-026-567893_OP4083409

**Clave
Primaria
ticket**

FastAPI: parseo de tickets



Diccionario

```
jsonTicket = {
    "_id": clauPrimaria,
    "idUsuari": idUsuari_enToken,
    "productesAdquirits": diccProductes,
    "totalTicket": preuTotalTicket,
    "direccioSuper": direccioSuper,
    "data": data_ISO8601,
    "hora" : hora
}
print(json.dumps(jsonTicket, indent=4, ensure_ascii=False))
```



```
{  
    "_id": "2423-026-567893_0P4083409",  
    "idUsuari": 3,  
    "productesAdquirits": {},  
    "totalTicket": 33.36,  
    "direccioSuper": "C/ SENYERA 24 46006 VALENCIA",  
    "data": "2025-04-30",  
    "hora": "21:14"  
}
```

FastAPI: parseo de tickets



Diccionario

```
jsonTicket = {  
    "_id": clauPrimaria,  
    "idUsuari": idUsuari_enToken,  
    "productesAdquirits": diccProductes,  
    "totalTicket": preuTotalTicket,  
    "direccioSuper": direccioSuper,  
    "data": data_IS08601,  
    "hora": hora  
}  
print(json.dumps(jsonTicket, indent=4, ensure_ascii=False))
```



```
{  
    "_id": "2423-026-567893_0P4083409",  
    "idUsuari": 3,  
    "productesAdquirits": {}, ←  
    "totalTicket": 33.36,  
    "direccioSuper": "C/ SENYERA 24 46006 VALENCIA",  
    "data": "2025-04-30",  
    "hora": "21:14"  
}
```

FastAPI: parseo de tickets

Delimitamos la tabla de productos (gracias a la cabecera)

07/01/2025 19:21 OP: 3322341

FACTURA SIMPLIFICADA: 3960-022-157614



Descripció	P. Unit	Import
2 12 OUS GRANS L	2,28	4,56
1 BOSSA PLÀSTIC		0,15
1 TOMAQ. PERA TERRINA		2,20
1 PEBROT FREGIR 0,184 kg	2,39 €/kg	0,44
TOTAL (€)		7,35
TARGETA BANCÀRIA		7,35

```
capsalTaulaProducts = ll_liniesTicket[6] #Descripción P.  
if not "Descripció" in capsalTaulaProducts:  
    raise ValueError("No se encontró la cabecera Descripción")  
  
taulaProducts = ll_liniesTicket[7:]  
for i in range(len(taulaProducts)):  
    print(taulaProducts[i])
```

```
212 OUS GRANS L 2,28 4,56  
1BOSSA PLÀSTIC 0,15  
1TOMÀQ. PERA TERRINA 2,20  
1PEBROT FREGIR  
0,184 kg 2,39 €/kg 0,44
```

Delimitador

Figure: Este proceso depende de encontrar la cabecera en la línea siete (funciona para catalán y castellano). Lanzamos excepcion si falla.

FastAPI: parseo de tickets

1a Detección productos envasados (No granel)

07/01/2025 19:21 OP: 3322341

FACTURA SIMPLIFICADA: 3960-022-157614



Descripció	P. Unit	Import
2 12 OUS GRANS L	2,28	4,56
1 BOSSA PLÀSTIC		0,15
1 TOMAQ. PERA TERRINA		2,20
1 PEBROT FREGIR		
0,184 kg	2,39 €/kg	0,44
	TOTAL (€)	7,35
	TARGETA BANCÀRIA	7,35

Primeras detección
- Ocupan solo una línea
- Verificados por 4 caracteres finales → RegEx (d,dd)

Productos que NO son a granel



Figure: Procedimiento: primera aproximación a la detección de productos que no son a granel mediante su importe.

FastAPI: parseo de tickets

1a Detección productos a granel

07/01/2025 19:21 OP: 3322341

FACTURA SIMPLIFICADA: 3960-022-157614



Descripció	P. Unit	Import
2 12 OUS GRANS L	2,28	4,56
1 BOSSA PLÀSTIC		0,15
1 TOMÀQ. PERA TERRINA		2,20
1 PEBROT FREGIR 0,184kg	2,39€/kg	0,44
	TOTAL (€)	7,35
	TARGETA BANCÀRIA	7,35

Primera
detección ←

- Ocupan dos líneas
- No tienen Importe en línea inicial



Productos que **SÍ** son a granel



Figure: Procedimiento: primera aproximación a la detección de productos a granel a partir de la falta de importe en su primera línea.

1a Detección productos envasados (continuación)

\forall producto envasado \exists “d,dd” al final de línea (importe).

- Si se compra → **una unidad**, entonces:
 - **No** Existe patrón “d,dd” a su izquierda (columna P.Unit)?
 - Si se compran → **2 o más unidades**, entonces:
 - **Sí** Existe patrón “d,dd” a su izquierda (columna P.Unit)?

FastAPI: parseo de tickets

07/01/2025 19:21 OP: 3322341		
FACTURA SIMPLIFICADA: 3960-022-157614		
		
Descripció	P. Unit	Import
2 12 OUS GRANS L	2,28	4,56
1 BOSSA PLÀSTIC		0,15
1 TOMÀQ. PERA TERRINA		2,20
1 PEBROT FREGIR 0,184 kg	2,39 €/kg	0,44
		TOTAL (€) 7,35
		TARGETA BANCÀRIA 7,35

Se pierde
el espacio



07/01/2025
212 OUS GRANS L 2,28 4,56
1BOSSA PLÀSTIC 0,15
1TOMÀQ. PERA TERRINA 2,20
1PEBROT FREGIR
0,184 kg 2,39 €/kg 0,44



Figure: **producto conflictivo envasado**: número de unidades queda mezclado con el inicio de la descripción o nombre de un producto imposibilitando segmentar ambos datos mediante espacio (*split()*)

FastAPI: parseo de tickets

07/01/2025 19:21 OP: 3322341



Solución

Se pierde el espacio

$$2 = \frac{4,56}{3,39}$$

12 OUS GRANS L 2,28 4,56
1BOSSA PLÀSTIC 0,15
1TOMÀQ. PERA TERRINA 2,20
1PEBROT FREGIR
0,184 kg 2,39 €/kg 0,44

```
quantitat = round(importProducte/preuUnitari)
nreDigits_PerLesquerra_Aesborrar = len(str(quantitat)) → tomamos longitud de
nomProducte = " ".join(l1.linhaP[-2])nreDigits_PerLesquerra_Aesborrar: 2.28
```

Figure: Solución al conflicto: se calcula qué parte de los dígitos pertenecen al número de unidades adquiridas y qué parte al nombre o descripción del mismo mediante cociente Importe/precioUnitario

FastAPI: parseo de tickets



MERCADONA, S.A. A-46103834
AVDA. ALCALDE ROVIRA ROURE 87
25198 LLEIDA
TELÉFON: 973305750
18/12/2024 12:58 OP: 4003395
TURIA SIMPELLEICITADA: 4037-010-778810



Descripció	P. Unit	Import
1 PAPATÀ 3 KG		5,00
1 CORS CABDELLS		1,89
1 ESPINACS		1,22
1 BOSSA PLÀSTIC		0,15
1 FILM TRANSPARENT		1,85
1 PIT FAM.		5,52
1 12 OUS PAGÈS		2,67
1 KIWI GROC SAFETA		3,51
1 PORRO TALLAT		2,86
1 PERA ROCHE 0,514 kg	1,99 €/kg	1,02
1 CARBASSÓ VERD 0,668 kg	1,80 €/kg	1,20
1 ALVOCAT 0,460 kg	5,50 €/kg	2,53
1 PARQUING ENTRADA 12:37	SORTIDA 12:58	0,00
TOTAL (€)		29,42

Al no encontrar patrón {d,dd} en la segunda línea, se confunde con la primera línea de un producto a granel.

Por lo tanto se tratará de ver la siguiente línea para encontrar kg, €/kg, etc...

Dado que no hay más líneas aquí dará IndexError

Figure: producto conflictivo: Sale un parking que podemos confundir por un producto envasado (primera línea) y uno a granel (2a línea) que no tendría la línea que lo suele seguir con los datos a extraer.

FastAPI: parseo de tickets



MERCADONA, S.A. A-46103834

AVDA. ALCALDE ROVIRA ROURE 87
25198 LLEIDA
TELÉFON: 973305750
18/12/2024 12:58 OP: 4003395
CTUR A SIMPLIFICADA 4037 010 7798



Descripció	P. Unit	Import
1 PATATA 3 KG		5,00
1 CORS CABDELLS		1,89
1 ESPINACS		1,22
1 BOSSA PLÀSTIC		0,15
1 FILM TRANSPARENT		1,85
1 PIT FAM.		5,52
1 12 OUS PAGÈS		2,67
1 KIWI GROC SAFETA		3,51
1 PORRO TALLAT		2,86
1 PERA ROCHA 0,514 kg	1,99 €/kg	1,02
1 CARBASSÓ VERD 0,668 kg	1,80 €/kg	1,20
1 ALVOCAT 0,460 kg	5,50 €/kg	2,53
1 PARQUING		0,00
ENTRADA 12:37	SORTIDA 12:58	
	TOTAL (€)	29,42

```
t = 0
diccProductos = {}
while t < len(taulaProductes):
    lineaP = taulaProductes[t]
    if "PARQUEO" in lineaP:
        t = t + 1
    elif esUnPreul(lineaP[-4:]):
        esGranel = False
        #
        #
        # procesado y guardado a dicc
    else:
        esGranel = True
        #
        #
        # procesado y guardado a dicc
```

2 saltos

Figure: Solución al conflicto: Saltamos la línea que contiene "PARKING" y la siguiente sin llegar a procesar nada de su contenido: no es de interés.

FastAPI: parseo de tickets

25/06/2024 12:01 OP: 547954



Descripción	P. Unit	Importe
1 BRONCHALES 6L		1,26
1 BRONCHALES 1,5L		0,39
1 LECHE SEMI CALCIO		0,96
1 BOLSA PLASTICO		0,15
1 PISTACHO TOSTADO SAL		3,35
1 12 HUEVOS SUPER GRAN		3,25
1 CORAZONES COGOLLO		1,98
1 FRESA		3,34
PESCADOS		
DORADA		
0,460 kg	8,75 €/kg	4,03
1 BANANA		
0,366 kg	1,45 €/kg	0,53
TOTAL (€)		19,24

Figure: producto conflictivo a granel: El producto ocupa tres líneas en vez de dos. El conflicto viene por partida doble: se añade una línea por encima con la categoría y esta primera línea -y la segunda- NO tiene un número “1” de unidades como en el resto de productos a granel.

FastAPI: parseo de tickets

25/06/2024 12:01 OP: 547954



Descripción	P. Unit	Importe
1 BRONCHALES 6L		1,26
1 BRONCHALES 1,5L		0,39
1 LECHE SEMI CALCIO		0,96
1 BOLSA PLASTICO		0,15
1 PISTACHO TOSTADO SAL		3,35
1 12 HUEVOS SUPER GRAN		3,25
1 CORAZONES COGOLLO		1,98
1 FRESA		3,34
PESCADOS		
DORADA		
0,460 kg	8,75 €/kg	4,03
1 BANANA		
0,366 kg	1,45 €/kg	0,53
TOTAL (€)		19,24



Figure: Solución al conflicto: No la vamos a implementar por ahora,
porque queremos forzar que salgan errores, tal y como sería para la
aplicación en producción (tickets no vistos previamente, casos imposibles
de prever sin un enfoque empírico)

FastAPI: parseo de tickets

Paso 3: Inicia la extracción de datos.

Una vez conforme con los tickets subidos en el paso anterior, clica en el engranaje: entonces extraeremos el contenido de los tickets digitales en PDF y analizaremos sus datos. Serás redirigido a tu *dashboard de visualización* al terminar.



extraídos de 285 tickets facilitados:
(7% de éxito)

Estos tickets no pudieron ser extraídos. No se analizarán.

- 20240625 Mercadona 19,24 €.pdf
- 20240706 Mercadona 22,07 €.pdf
- 20240712 Mercadona 15,72 €.pdf
- 20241005 Mercadona 12,82 €.pdf
- 20241205 Mercadona 26,62 €.pdf
- 20241214 Mercadona 45,61 €.pdf
- 20250213 Mercadona 25,43 €.pdf
- 20250221 Mercadona 14,22 €.pdf

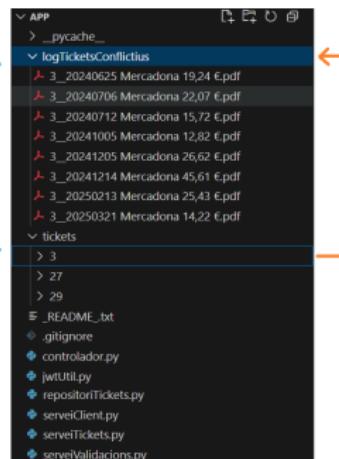


Figure: Así mostraremos los tickets cuyo parseo ha fallado al usuario y los guardaremos a parte concatenando el id de usuario y el ticket

FastAPI: parseo de tickets

C/ SENYERA 24
46006 VALENCIA
TELÉFONO: 963341702
07/02/2025 12:45 OP: 468149
FACTURA SIMPLIFICADA: 2423-022-563148



Descripción	P. Unit	Importe
1 PATATA 3 KG		5,10
2 12 HUEVOS GRANDES-L	2,28	4,56
1 POLLO ENTERO LIMPIO		6,44
1 AGUACATE BANDEJA		3,49
1 TOMATE PERA TARRINA		2,37
1 MANDARINA 0,264 kg	2,19 €/kg	0,58
1 BANANA 0,310 kg	1,52 €/kg	0,47
TOTAL (€)		23,01



Figure: Ticket parseado correctamente a su formato JSON (persistible)

FastAPI: parseo de tickets

Persistencia | FastAPI → mongoDB

MongoDB (NoSQL)	MySQL (SQL)
Colección (tickets)	Tabla
Documento JSON → BSON (ticket)	Fila
Campos anidados: esquema flexible	Campos fijos: esquema rígido

Table: Comparación entre MongoDB y MySQL

FastAPI: parseo de tickets

Persistencia | FastAPI → mongoDB

Se ha escogido MongoDB porque es una base de datos noSQL, ideal en el caso que aquí nos ocupa:

- No existe un número **fijo** de productos en un ticket.
 - **No tenemos** un listado exhaustivo de todos los productos del supermercado.
 - En el “*inflalyzer*” queremos hacer **filtros por productos que todavía no conocemos**.

FastAPI: parseo de tickets

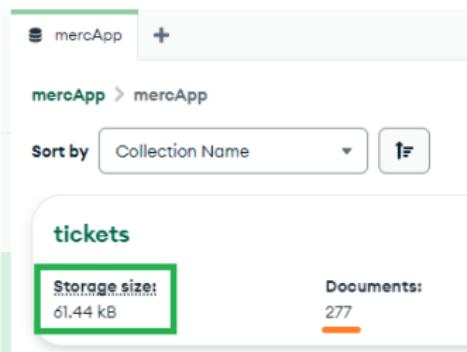
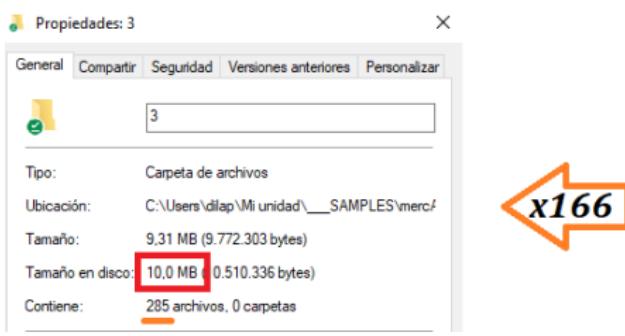


Figure: Guardar los tickets en formato BSON tiene una implicación evidente más allá de la posibilidad de hacer búsquedas.

FastAPI: exposición tickets (+ dashboard)

```
$ python repositoryTickets.py
{
    "BOLSA PLÁSTICO": 132,
    "BANANA": 53,
    "BRONCHALES 6L": 58,
    "BRONCHALES 1,5L": 58,
    "12 HUEVOS GRANDES": 1,
    "TOMATE PERA TARRINA": 37,
    "AGUACATE BANDEJA": 31,
    "LECHE SEMI+": 25,
    "CORIZONES COGOLLO": 24,
    "LS POLLO REBOZADO": 23,
    "CALABACIN VERDE": 23,
    "SURTIDO DULCES": 22,
    "T. SETAS": 20,
    "PUEBRO CORTADO": 28,
    "BEZOVA 1,5L": 19,
    "JAKON S. EXTRAFINO": 19,
    "PAN DE MASA": 18,
    "BRONCHALES 6X1,5L": 17,
    "CHULETA PAVO": 16,
    "BOSSA PLÁSTIC": 15,
    "KEFIR": 15,
    "FRESÓN": 15,
    "NECTARINA": 15,
    "MANDARINA": 14,
```

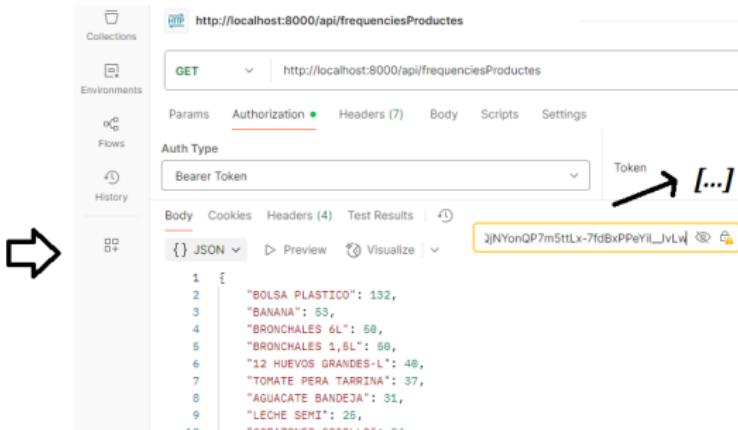


Figure: Extracción de datos de un endpoint de fastAPI desde su salida del repositorioTickets.py (donde hacemos la consulta en MongoDB) hasta su llamada a controlador.py a través de POSTMAN mediante token de acceso (permisos a 1)

FastAPI: exposición tickets (+ dashboard)

```
//NOMBRE PRODUCTES DIFERENTES ADQUIRITS
fetch('http://localhost:8000/api/frequenciesProductes', {
  method: "GET",
  headers: {
    'Authorization': "Bearer " + localStorage.getItem("AccessToken"),
    'Accept': 'application/json'
  }
})
.then(response => {
  if (!response.ok) {
    throw new Error('La solicitud devolvió código de error en paso3 : ' + response.status)
  }
  return response.json();
})
.then(diccDades => { //estil --> {'BOLSA PLASTICO': 132, 'BANANA': 53, 'BRONCHALES 6L': 5
  const dom_nreProdDiferents = document.getElementById("nreProductesDiferentsAdquirits")
  dom_nreProdDiferents.innerHTML = Object.entries(diccDades).length;
})
.catch(error => {
  console.error('Error en paso3:', error);
});
```

Figure: En lugar de postman ahora lo sacamos hacia el front con la función Fetch de JavaScript. Sacamos números de pares clave:valor del JSON (Nótese: Python (*dict*) → JSON → JavaScript(*Object*))

FastAPI: exposición tickets (+ dashboard)

```
//NOMBRE PRODUCTES DIFERENTS ADQUIRITS
fetch('http://localhost:8000/api/frequenciesProductes', {
    method: "GET",
    headers: {
        'Authorization': "Bearer " + localStorage.getItem("AccessToken"),
        'Accept': 'application/json'
    }
})
.then(response => {
    if (!response.ok) {
        throw new Error('La solicitud devolvió código de error en paso3 : ' + response.status)
    }
    return response.json();
})
.then(diccDades => { //estil --> {'BOLSA PLASTICO': 132, 'BANANA': 53, 'BRONCHALES 6L': 5
    const dom_nreProdDiferents = document.getElementById("nreProductesDiferentsAdquirits")
    dom_nreProdDiferents.innerHTML = Object.entries(diccDades).length;
    localStorage.setItem("frequenciesProductes", Object.entries(diccDades)); //guardem les freqs
})
.catch(error => {
    console.error('Error en paso3:', error);
});
```

Figure: Persistimos los datos en el localStorage del cliente. Volveremos a usarlos después en el “inflalyzer” y así ahorraremos sucesivas llamadas a BBDD para unos datos que ya se extrajeron una vez

FastAPI: exposición tickets (+ dashboard)

POST http://localhost:8000/api/graficDataPreuProducte

Params Authorization • Headers (9) Body • Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 {"nomProducte": "POLLO ENTERO LIMPIO"}

Body Cookies Headers (4) Test Results

{ } JSON Preview Visualize

```
1 [  
2 {  
3   "x": "2024-05-30",  
4   "y": 5.81  
5 },  
6 {  
7   "x": "2024-07-17",  
8   "y": 5.97  
9 },
```

Figure: Endpoint que permite hacer el gráfico de inflación de un producto, obteniendo sus puntos X (fecha aaaa-mm-dd) e Y (€/kg o €/unidad)

FastAPI: exposición tickets (+ dashboard)

http://localhost:8000/api/gastosPerCategoria

GET http://localhost:8000/api/gastosPerCategoria

Params Authorization Headers (7) Body Scripts Settings

Auth Type Bearer Token Token

Body Cookies Headers (4) Test Results ⚙️

{ } JSON ▾ Preview Visualize ▾

```
1 {  
2   "1": 598.3100000000001,  
3   "2": 399.44,  
4   "3": 460.55,  
5   "4": 249.8,  
6   "5": 67.74,  
7   "6": 490.57,  
8   "7": 115.7,  
9   "8": 287.95,  
10  "9": 144.92,  
11  "10": 60.5,  
12  "11": 235.67,  
13  "12": 5.95,  
14  "13": 53.44  
15 }
```

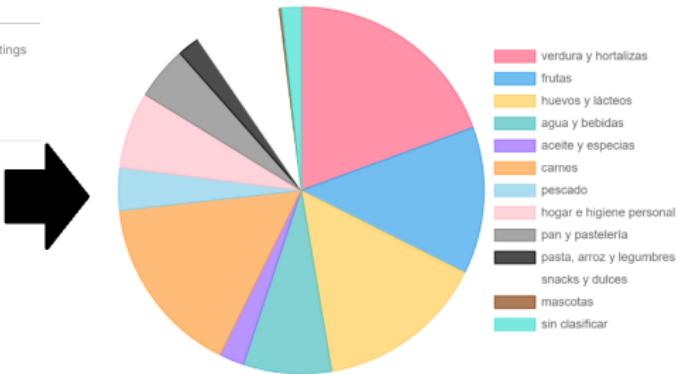


Figure: Endpoint utilizado para obtener los gastos totales por cada una de las 13 categorías de alimentación.

Front-end: Vanilla JS

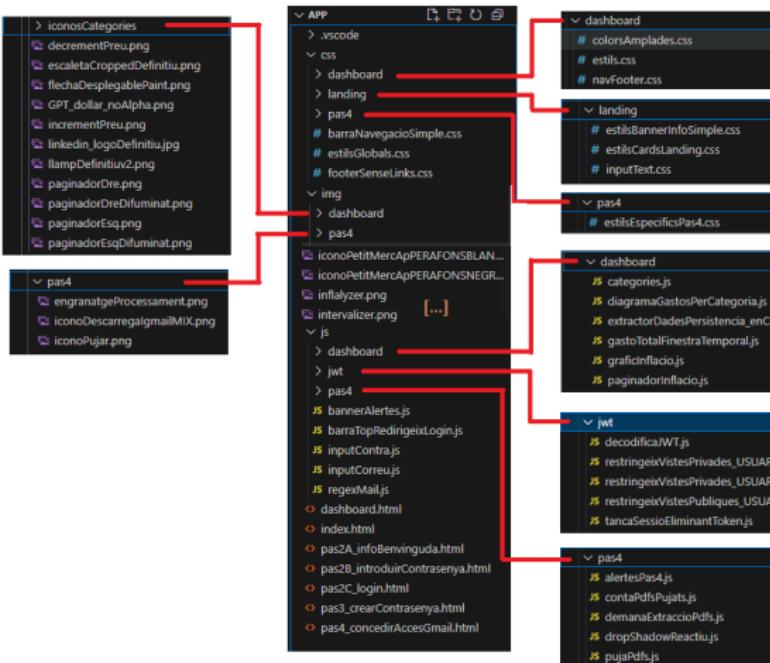


Figure:

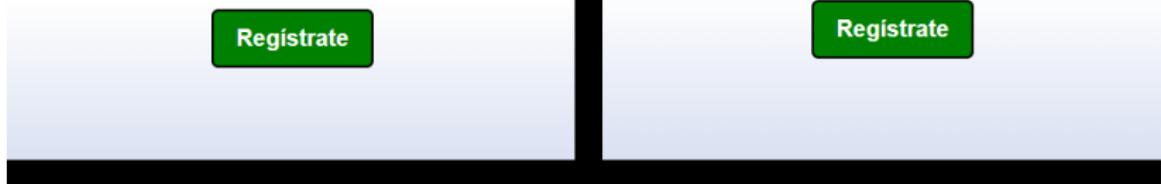
Front-end: Vanilla JS

Dirección de correo →

Register

Dirección de correo _____

[Regístrate](#)



Dirección de correo _____
asd

¡El formato del correo no es correcto!

Registrazione

Dirección de correo

asd@gmail.com





Front-end: Vanilla JS

M

Iniciar sesión

The image shows the Netflix account setup screen. At the top left is the red 'NETFLIX' logo. To its right is a small blue 'Iniciar sesión' button. The main area features three icons representing a laptop, a desktop monitor, and a smartphone, all outlined in red. Below these icons is a large red button with the text 'Siguiente'. Above the button, the text reads: 'Termina de configurar tu cuenta' (Finish setting up your account) and 'Netflix personalizado para ti. Crea una contraseña para empezar a ver Netflix.' (Personalized Netflix for you. Create a password to start watching Netflix.).

Front-end: Vanilla JS

Demostración!

Contenidos

1 Introducción

2 Diseño

- Requisitos
 - Diagramas de sistemas

3 Desarrollo

- Entornos de desarrollo
 - Despliegue
 - Spring Boot: gestión usuarios
 - FastAPI: parseo de tickets
 - FastAPI: exposición tickets (+ dashboard)
 - Front-end: Vanilla JS

4 Conclusiones

Conclusiones

- Definir y desarrollar dos back-ends desde cero, uno para usuarios y otro para tickets digitales, integrándolos entre sí.
- Aprender y manejar bases de datos NoSQL (MongoDB) y relacionales (MySQL), junto con el mapeo de objetos Java a bases relacionales usando JPA.
- Entender y programar la gestión de autenticación y autorización con JWT en Java, creando clases personalizadas para generar y validar tokens.
- Programar el front-end con JavaScript puro, controlando vistas según tokens de acceso y aplicar CSS desde cero, incluyendo media queries sin frameworks.
- Contenerizar tanto back-end como front-end y automatizar despliegues. Configurar CORS.

Introducción

Diseño
○
○○○
○○○

A decorative graphic consisting of a grid of small white circles arranged in a pattern that tapers to the right.

Conclusiones

Gracias por vuestra atención

¿Preguntas?