

Ampliació del projecte RA8 i RA9 - Compra i venda de Cotxes

TAULA DE CONTINGUTS

TAULA DE CONTINGUTS.....	1
INTRODUCCIÓ.....	2
MODIFICACIONS DEMANADES.....	2
COM M'HE TROBAT EL CODI REBUT?.....	3
COM HE IMPLEMENTAT L'ÚS DE BASES DE LES BASES DE DADES.....	3
QUÈ HE APROFITAT D'ALTRES ASSIGNATURES?.....	6
COM S'HA REPARTIT I ORGANITZAT LA FEINA?.....	7
COM FUNCIONA ARA L'APLICACIÓ AMB LES MODIFICACIONS.....	8
COM FUNCIONA L'APP AMB LES CONSULTES A LES BBDDs?.....	18
EXTENSIONS DE L'APLICACIÓ.....	30

INTRODUCCIÓ

En aquest projecte es demana modificar un programa senzill per practicar la gestió de dades, en aquest cas s'havia de passar d'utilitzar arxius JSON a gestionar les dades amb Bases de dades relacionals i documentals, i per fer-ho hauria de dedicar 1 mes en treballar amb aquest programa, i com sabia que durant aquest temps li agafaria carinyo el vaig batejar com a **TradeMotor**.

Versió de python utilitzada: *Python 3.13.3*

MODIFICACIONS DEMANADES

En aquest treball la principal modificació que es demanava era canviar el sistema de gestió de dades, a més d'utilitzar documents dins del codi que guarden les dades havíem d'implementar dues bases de dades, **MariaDB** (relacional) i **Mongodb** (documental).

A pesar de trobar diferents errors durant el procés per temes d'incompatibilitat amb les llibreries al final es va poder integrar utilitzant **pymysql** i **pymongo**.

També es demanava crear tot un sistema de funcionament:

1. **Un venedor publica un cotxe** ✓
 - Es guarda a **MariaDB** (**cotxes**). ✓
2. **Un comprador guarda el cotxe com a favorit** ✓
 - Es guarda a **MongoDB** (**favorits**). ✓
3. **Un comprador fa una oferta pel cotxe** ✓
 - Es guarda a **MongoDB** (**ofertes**). ✓
4. **El venedor accepta l'oferta i es ven el cotxe** ✓
 - Es mou de **MongoDB** (**ofertes**) a **MariaDB** (**transaccions**). ✓
5. **El comprador escriu una ressenya del cotxe** ✓
 - Es guarda a **MongoDB** (**comentaris**). ✓
6. **Es mostra l'historial de preus del cotxe** ✓
 - Es recupera de **MongoDB** (**historial_preus**). ✓

Després de moltes hores de treball i recerca de funcionament i solució d'errors es van poder integrar totes les modificacions.

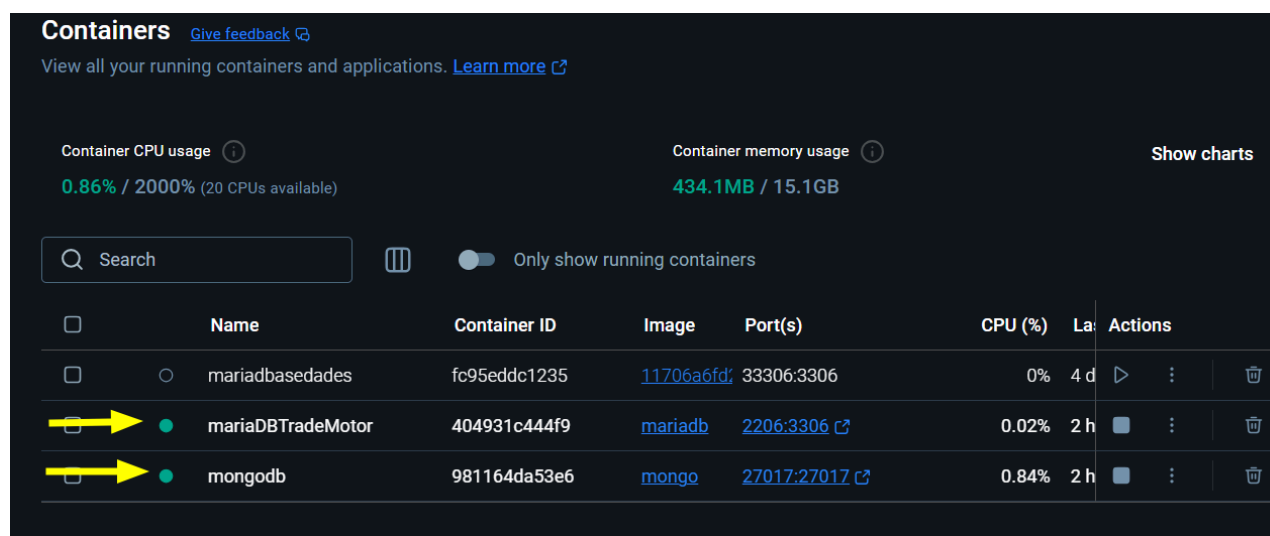
COM M'HE TROBAT EL CODI REBUT?

El codi que vaig rebre tenia una estructura senzilla, els arxius s'emmagatzemaven amb fitxers JSON dins del propi codi i tenia tres funcions, publicar un anunci, veure l'anunci i un filtre de búsqueda. Les dades eren les bàsiques d'un cotxe: tipus, marca, model, any i preu. El programa estava en castellà així que vaig decidir continuar-lo en aquest idioma. El codi utilitzava HTML, CSS per al frontend i PYTHON per al backend amb flask.

COM HE IMPLEMENTAT L'ÚS DE BASES DE LES BASES DE DADES

Per gestionar les bases de dades vaig optar per emmagatzemar els servidors en contenidors amb docker, el que millora la gestió d'aquestes al tenir un entorn controlat per connectar-les, desconnectar-les i accedir-hi per modificar-les.

ADMINISTRACIÓ DE LES BBDD

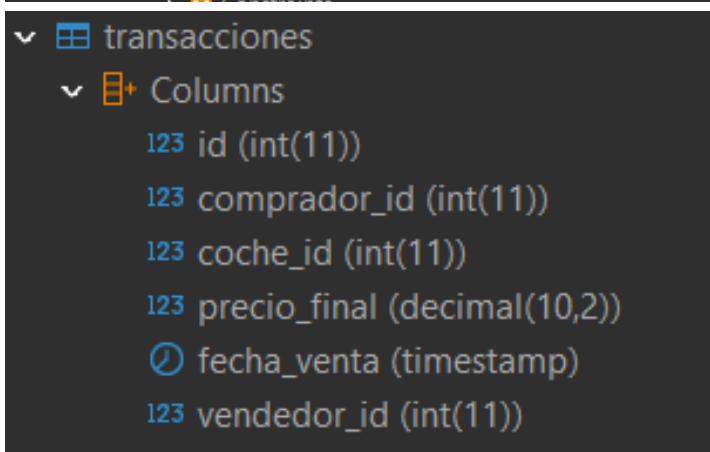
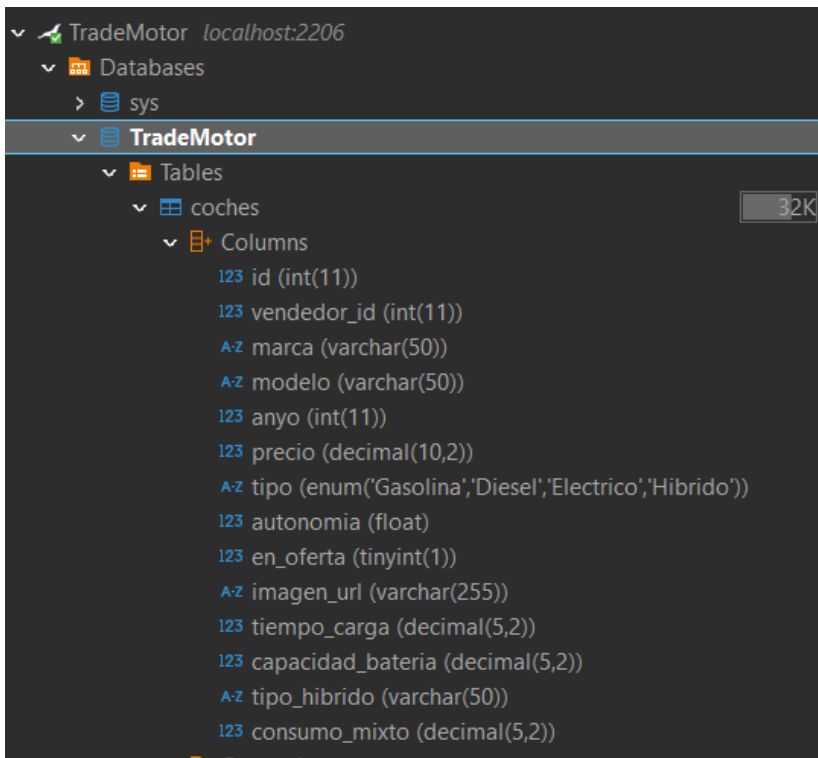


The screenshot shows the Docker Desktop interface. At the top, it says 'Containers' with a link to 'Give feedback'. Below that, it says 'View all your running containers and applications. Learn more'. There are two status bars: 'Container CPU usage' at 0.86% / 2000% (20 CPUs available) and 'Container memory usage' at 434.1MB / 15.1GB. A 'Show charts' button is on the right. Below the status bars is a search bar and a toggle switch for 'Only show running containers'. The main part of the screen is a table of containers.

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	La	Actions
<input type="checkbox"/>	mariadbasedades	fc95eddc1235	11706a6fd	33306:3306	0%	4 d	
<input checked="" type="checkbox"/>	mariaDBTradeMotor	404931c444f9	mariaadb	2206:3306	0.02%	2 h	
<input checked="" type="checkbox"/>	mongodb	981164da53e6	mongo	27017:27017	0.84%	2 h	

MARIADB

```
PS C:\Users\velcat> docker run --name mariaDBTradeMotor -e MYSQL_ROOT_PASSWORD=iker -e MYSQL_USE
R=velcat -e MYSQL_PASSWORD=velcat -p 2206:3306 -d mariadb
Unable to find image 'mariadb:latest' locally
latest: Pulling from library/mariadb
0c9f57c1bb30: Download complete
2b221cf763a8: Download complete
5e4180757702: Download complete
0b86886c6aaa: Download complete
2726e237d1a3: Download complete
bbef7eafa75b: Download complete
43028b9f5f8e: Download complete
ab732728101f: Download complete
Digest: sha256:81e893032978c4bf8ad43710b7a979774ed90787fa32d199162148ce28fe3b76
Status: Downloaded newer image for mariadb:latest
404931c444f959b8656f4baaef3c2f07e70f8c02e022a350f235541c61aaee72
```



```
▼ usuarios
  ▼ Columns
    123 id (int(11))
    A-Z nombre (varchar(100))
    A-Z email (varchar(100))
    A-Z tipo_usuario (varchar(20))
    A-Z password (varchar(255))
```

MONGODB

The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS (2)' panel lists two connections: 'cluster0.q14yb.mongodb.net' and 'localhost:27017'. The 'localhost:27017' connection is selected, and its database structure is shown below it, including 'TradeMotor' (which contains collections 'comentarios', 'favoritos', 'historial_precios', and 'ofertas') and other databases like 'admin', 'config', 'local', 'tasques_db', and 'test'. On the right, the 'Database Statistics' panel shows details for four collections: 'comentarios', 'favoritos', 'historial_precios', and 'ofertas'. Each collection's statistics are as follows:

Collection	Storage size	Documents	Avg. document size	Indexes	Total index size
comentarios	8.19 kB	0	0 B	1	12.29 kB
favoritos	8.19 kB	0	0 B	1	12.29 kB
historial_precios	20.48 kB	4	92.00 B	1	36.86 kB
ofertas	8.19 kB	0	0 B	1	12.29 kB

Un cop tenia les bases de dades preparades vaig utilitzar llibreries de python per crear les connexions amb el programa.

```
import pymysql
from pymongo import MongoClient
```

```
# Mariadb
def get_mariadb_connection():
    return pymysql.connect(
        host='192.168.221.149',
        port=2206,
        user='root',
        password='iker',
        database='TradeMotor'
    )
```

```
# MongoDB
mongo_client = MongoClient('mongodb://admin:adminpassword@192.168.221.149:27017/')
mongo_db = mongo_client['TradeMotor']
comentarios_col = mongo_db['comentarios']
favoritos_col = mongo_db['favoritos']
ofertas_col = mongo_db['ofertas']
historial_precios_col = mongo_db['historial_precios']
```

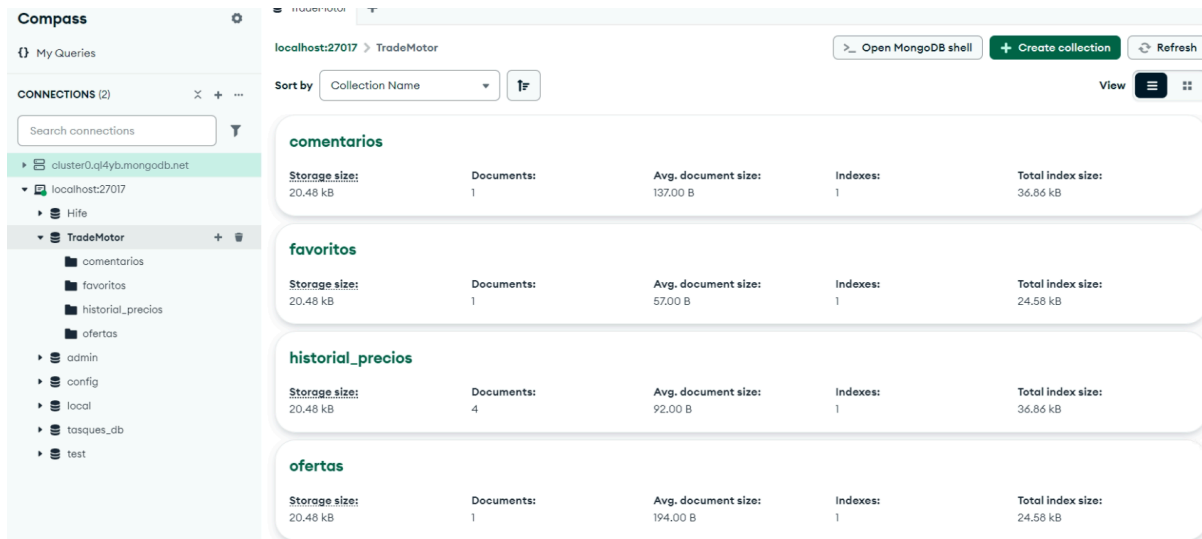
A partir d'aquí utilitzava funcions per introduir i extreure les dades de la base de dades i mostrarles a la meua pàgina web.

```
conn = get_mariadb_connection()
with conn.cursor() as cursor:
    cursor.execute(query, tuple(params))
    rows = cursor.fetchall()
conn.close()

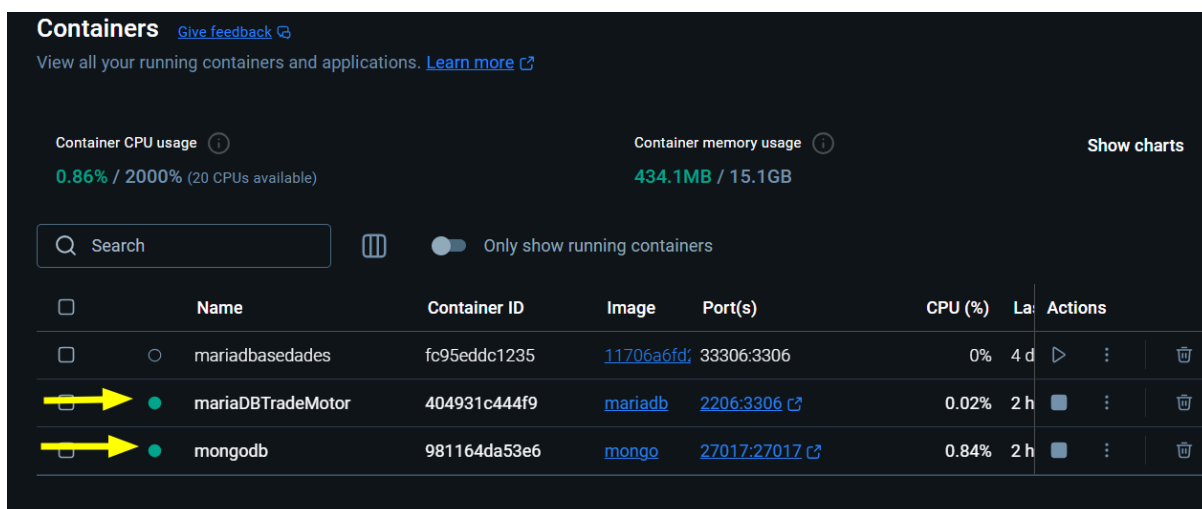
coches = [
    Coche(id=row[0], marca=row[1], modelo=row[2], precio=row[3], anyo=row[4], tipo=row[5],
          en_oferta=bool(row[7]), tiempo_carga=row[8], capacidad_bateria=row[9],
          tipo_hibrido=row[10], consumo_mixto=row[11], imagen_url=row[12])
    for row in rows
]
```

QUÈ HE APROFITAT D'ALTRES ASSIGNATURES?

M'he estalviat temps aprenent a configurar una base de dades documental ja que a mòdul optatiu vam estar aprenent a utilitzar mongodb.



També he aprofitat que a mòdul optatiu vam estar utilitzant docker per acabar d'aprendre a utilitzar-lo i poder gestionar les meves bases de dades amb contenidors i que jo tenia coneixements sobre com preparar una base de dades de mariadb d'altres cursos.



COM S'HA REPARTIT I ORGANITZAT LA FEINA?

Com en aquest projecte volia carregar amb totes les tasques per aprendre a fer-ho tot jo m'he tingut que organitzar de manera optima per finalitzar a temps.

En primer lloc vaig dedicar un temps a pensar que és el que havia de fer i com fer-ho. Un cop tenia planejat com anar construint el programa vaig passar a l'acció i vaig anar muntant pàgina per pàgina el programa, construïa la base del html, preparava la funció i després acabava

d'implementar aquesta funció al html per anar corregint errors ja que a vegades em trobava amb errors d'escriptura que em poden fer perdre molt de temps.

```
templates > index.html > ...
1  <!DOCTYPE html>
2  <html lang="ca">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Inicio - TradeMotor</title>
7      <link rel="stylesheet" href="../static/inicio.css">
8      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.6/dist/css/bootstrap.min.css" rel="stylesheet">
9  </head>
10 <body>
11     <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top w-100">
12         <div class="container-fluid">
13             <a class="navbar-brand" href="{{ url_for('index') }}">TradeMotor</a>
14             <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
15                 data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
16                 aria-label="Toggle navigation">
17                 <span class="navbar-toggler-icon"></span>
18             </button>
19         </div>
20     </nav>
```

```
@app.route("/")
def index():
    return render_template("index.html", active_page="index")

def obtener_historial_precios(coche_id):
    historial = historial_precios_col.find_one({"coche_id": coche_id})
    return historial["precios"] if historial and "precios" in historial else []

def guardar_historial_precio(coche_id, nuevo_precio):
    result = historial_precios_col.update_one(
        {"coche_id": int(coche_id)},
        {"$push": {
            "precios": {
                "fecha": datetime.now().strftime("%Y-%m-%d"),
                "precio": nuevo_precio
            }
        }},
        upsert=True
    )
```

COM FUNCIONA ARA L'APLICACIÓ AMB LES MODIFICACIONS.

En primer lloc s'ha afegit un inici de sessió que és necessari per realitzar qualsevol interacció amb la base de dades, el que si es pot fer és veure els anuncis i les dades.

Registrarse Iniciar sesión

Lista de anuncios

Marca

Modelo

Precio máximo (€)

Año

Tipo



Filtrar

Volkswagen ID4 Pro

Tipo: electrico
Precio: 41000.0 €
Año: 2025



BMW Serie 3

Tipo: diesel
Precio: 49150.0 €
Año: 2023



Skoda Octavia

Tipo: diesel
Precio: 33400.0 €
Año: 2025

Superoferta 21% de descuento



Renault Clio

Tipo: diesel
Precio: 20529.0 €
Año: 2025



Dacia Sandero Essential

Tipo: gasolina
Precio: 12540.0 €
Año: 2025

Superoferta 21% de descuento



Ford Focus 1.0

Tipo: gasolina
Precio: 24500.0 €
Año: 2023



però si volem realitzar qualsevol interacció ens obligarà a iniciar sessió.

Volkswagen ID4 Pro

Tipo: electrico
Precio: 41000.0 €
Año: 2025

Añadir a favoritos



Inicia sesión

Correo electrónico:


Contraseña:

[Iniciar sesión](#)

¿No tienes una cuenta? [Regístrate aquí](#)

[Página principal](#)

Un cop iniciem sessió tindrem accés a tota la pàgina.

 Hola, Iker Vericat [Inicio](#) [Ventas](#) [Mis Compras](#) [Favoritos](#) [Cerrar sesión](#)

Mis favoritos

Volkswagen ID4 Pro

Tipo: electrico

Precio: 41000.0 €

Año: 2025

[✖ Eliminar](#) [👁 Hacer oferta](#)

PUBLICACIONES

Ara els venedors, després d'iniciar sessió poden accedir al botó "Publicar un anuncio", situat al inici, on se'ls desplega un formulari amb les diferents dades del vehicle. L'introducció de dades dependrà del tipus de vehicle.

Publicar un nuevo anuncio

Tipo de vehículo:

Diésel

Marca:

Modelo:

Precio:

Año:

¿Está en oferta? ☐ Sí

🖼️ Imágenes del coche:

Elegir archivos No se ha seleccionado ningún archivo

Publicar Anuncio

Els vehicles híbrids i elèctrics tenen paràmetres relacionats amb el seu tipus com es pot veure a continuació:

Tipo de vehículo:

Eléctrico

Autonomía (km):

Tiempo de carga (horas):

Capacidad de batería (kWh):

Marca:

Modelo:

Precio:

Año:

Publicar un nuevo anuncio

Tipo de vehículo:

Híbrido



Autonomía (km):

900

Tipo de híbrido:

HEV

Consumo mixto (L/100km):

4,5

Marca:

Kia

Modelo:

Niro


Precio:

25979

Año:

2025

¿Está en oferta? ☐ Sí

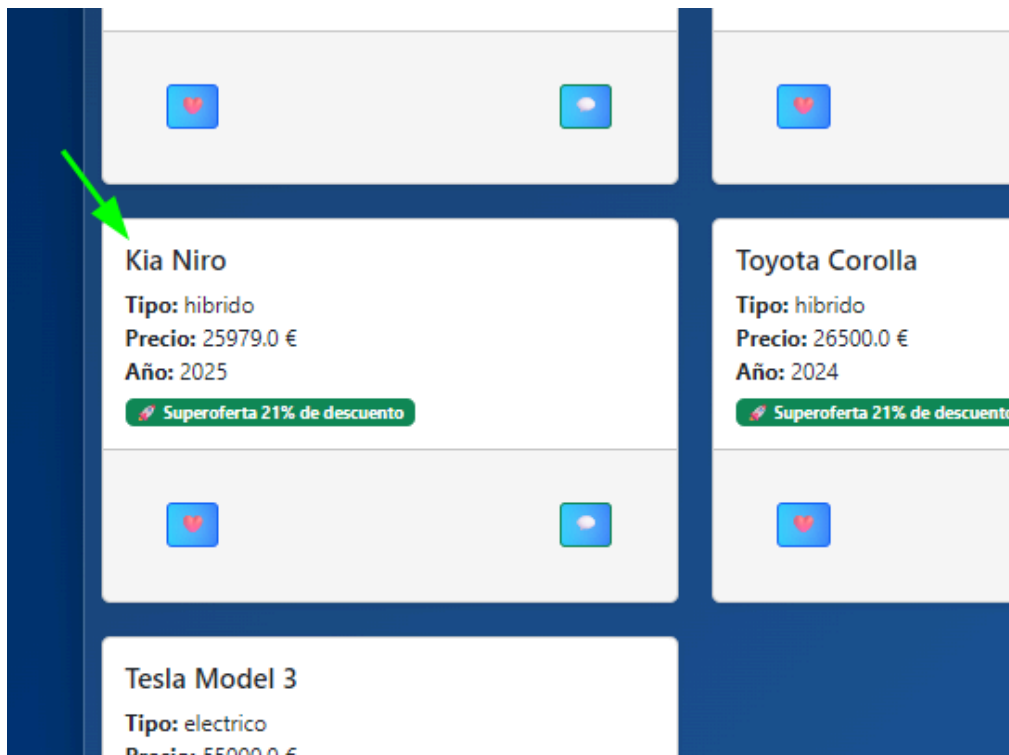
 Imágenes del coche:

Elegir archivos

No se ha seleccionado ningún archivo

Publicar Anuncio

I al fer clic a publicar aquest apareix a la pestanya “anuncis”.




Ara si un comprador vol comprar el cotxe s'ha de registrar amb el seu compte.

The screenshot shows a login form titled "Inicia sesión" on a dark blue background. It includes a label "Correo electrónico:" followed by a text input field containing "rodrigosalas@example.com". Below this is a label "Contraseña:" followed by a password input field with a toggle icon. A blue button labeled "Iniciar sesión" is positioned below the password field. At the bottom, there is a link "¿No tienes una cuenta?" followed by a button labeled "Regístrate aquí". A large blue button labeled "Página principal" is at the very bottom.



Al iniciar sessió podrà dirigir-se a l'anunci i guardarlo a favoritos o enviar una oferta al venedor.


Favorits:

Kia Niro
Tipo: híbrido
Precio: 25979.0 €
Año: 2025

 **Superoferta 21% de descuento**


Añadir a favoritos


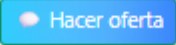
 



Mis favoritos


Kia Niro
Tipo: híbrido
Precio: 25979.0 €
Año: 2025


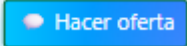
 **Superoferta 21% de descuento**


 

Oferta:

Kia Niro
Tipo: híbrido
Precio: 25979.0 €
Año: 2025

 **Superoferta 21% de descuento**



Haz una oferta

Monto de la oferta (€):

Mensaje al vendedor:

Un cop enviada la oferta al vendedor li apareix a la secció “Ventas” la oferta realitzada pel comprador on pot acceptar-la o rebutjar-la.

[✎ Editar](#)

[🗑 Eliminar](#)

Kia Niro

Tipo: híbrido

Precio: 25979.0 €

Año: 2025

🚀 Superoferta 21% de descuento

Ofertas de compradores:

Rodrigo Salas ofrece

25979.0 €

Acepto el precio del anuncio.

Aceptar
✓

Rechazar
✗

Al aceptar la compra es notifica al venedor amb un “flash” i el comprador rep la informació.

Vendedor:

Oferta aceptada y venta registrada correctamente.



Mis Coches en Venta

Comprador: Ara aquest pot posar comentaris sobre aquest vehicle per a que els usuaris puguin veure la seva opinió des dels anuncis.

Mis Compras

Kia Niro

Precio pagado: 25979.00 €

Fecha de compra: 2025-05-20 19:04:42

Escribe tu comentario aquí...

Enviar comentario

Kia Niro

Precio pagado: 25979.00 €

Fecha de compra: 2025-05-20 19:04:42

Me ha gustado mucho este vehiculo,
tiene buena autonomia y rendimiento
por el precio pagado 👍 |

Enviar comentario

Comentari del vehicle a "anuncios"

Kia Niro



Tipo: híbrido

Año: 2025

Precio: 25979.0 €

Autonomía: 900.0 km

Tipo de híbrido: HEV

Consumo mixto: 4.50 L/100km



Historial de precios:

No hay historial de precios disponible.

Comentarios:

Rodrigo Salas: Me ha gustado mucho este vehículo, tiene buena autonomía y rendimiento por el precio pagado 🍷. (2025-05-20 21:11:07.655000)

Cerrar

COM FUNCIONA L'APP AMB LES CONSULTES A LES BBDDs?

Com hem pogut veure abans, s'han realitzat moltes modificacions al codi, aquestes requereixen consultes a les bases de dades les quals mostraré a continuació:

REGISTRE I INICI DE SESSIÓ

Per a que un usuari es pugui registrar necessitem una funció que emmagatzema les dades introduïdes per aquest a un formulari i les envie a la base de dades, per fer-ho utilitzem la funció "registro" ubicada al [app.py](#).

```
@app.route("/registro", methods=["GET", "POST"])
def registro():
    if request.method == "POST":
        nombre = request.form["nombre"]
        email = request.form["email"]
        password = request.form["password"]
        tipo_usuario = request.form["tipo_usuario"]

        # Hash de la contraseña
        hashed_password = generate_password_hash(password)

        # Guardar el usuario en la base de datos
        conn = get_mariadb_connection()
        with conn.cursor() as cursor:
            cursor.execute(
                "INSERT INTO usuarios (nombre, email, tipo_usuario, password) VALUES (%s, %s, %s, %s)",
                (nombre, email, tipo_usuario, hashed_password)
            )
            conn.commit()
        conn.close()

        flash("Registro exitoso. Puedes iniciar sesión.")
        return redirect(url_for("login"))

    return render_template("registro.html")
```

Tables		Enter a SQL expression to filter results (use Ctrl+Space)				
> coches	32K	Grilla	123 id	Az nombre	Az email	Az tipo_usuario
> transacciones	48K					Az password
> usuarios	32K	Texto	1	Iker Vericat	ikervericat@iesmontsia.org	vendedor
> Columns			2	Alex Sans	alexsans@ejemplo.com	comprador
			3	admin	admin@example.com	vendedor

Aquesta a part de recullir les dades i guardarles a la taula usuaris a MariaDB encripta la contrasenya per a que aquesta no es pugui veure desde la base dades.

Ara per iniciar sessió s'utilitza una funció la qual recupera les dades dels camps corresponents verificant que aquests coincideixin, en cas contrari utilitza flash per avisar de que les credencials son incorrectes.

```
@app.route("/login", methods=["GET", "POST"])
def login():
    if request.method == "POST":
        email = request.form["email"]
        password = request.form["password"]
        conn = get_mariadb_connection()
        with conn.cursor() as cursor:
            cursor.execute("SELECT id, nombre, email, password FROM usuarios WHERE email = %s", (email,))
            usuario = cursor.fetchone()

            if usuario and check_password_hash(usuario[3], password):
                session["user_id"] = usuario[0]
                session["nombre"] = usuario[1]
                session["email"] = usuario[2]
                return redirect(url_for("index"))
            else:
                flash("Credenciales incorrectas.")
        return render_template("login.html")

@app.route("/logout")
def logout():
    session.clear()
    return redirect(url_for("index"))
```

PUBLICACIONES DEL VENEDOR

Per a que un venedor pugui realitzar una publicació ha d'estar registrat, això ho aconseguim amb la funció `@login_required`. Si aquest té accés escriurà les dades en un formulari i al enviar-ho una funció del backend guardarà les dades del usuari autenticat (venedor) i les dades del vehicle introduïdes al formulari.

```
@app.route("/publicar", methods=["GET", "POST"])
@login_required
def publicar():
    if request.method == "POST":
        # Capturar los datos del usuario autenticado
        nombre_vendedor = session.get("nombre")
        email_vendedor = session.get("email")

        if not nombre_vendedor or not email_vendedor:
            flash("Error: No se han encontrado los datos del vendedor.")
            return redirect(url_for("publicar"))

        # Capturar los datos del coche
        coche = Coche.from_form(request.form)
        tiempo_carga = request.form.get("tiempo_carga") if coche.tipo == "electrico" else None
        capacidad_bateria = request.form.get("capacidad_bateria") if coche.tipo == "electrico" else None
        tipo_hibrido = request.form.get("tipo_hibrido") if coche.tipo == "hibrido" else None
        consumo_mixto = request.form.get("consumo_mixto") if coche.tipo == "hibrido" else None
```

	id	vendedor_id	marca	modelo	anyo	precio	tipo	autonomia
1	33	1	Tesla	Model 3	2.025	55.000	Eléctrico	570
2	34	1	Hyundai	Ioniq 5 AWD	2.025	88.800	Eléctrico	500
3	36	1	Toyota	Corolla	2.024	26.500	Híbrido	900
4	37	1	Kia	Niro	2.025	25.979	Híbrido	900
5	38	1	Ford	Focus 1.0	2.023	24.500	Gasolina	[NULL]
6	39	1	Dacia	Sandero Essential	2.025	12.540	Gasolina	[NULL]
7	40	1	Renault	Clio	2.025	20.529	Diesel	[NULL]
8	47	1	Skoda	Octavia	2.025	33.400	Diesel	[NULL]
9	48	1	BMW	Serie 3	2.023	49.150	Diesel	[NULL]
10	50	1	Volkswagen	ID4 Pro	2.025	41.000	Eléctrico	950

Ara les dades del vehicle son recollides per la funció **anuncios** que les envia a **anuncios.html** i aquest les mostra per **pantalla**. En aquest html les dades bàsiques es mostraran dins d'una carta, a demés he implementat un modal que mostra totes les dades, però això ho explicaré posteriorment 😊.

```
@app.route('/anuncios', methods=['GET'])
def anuncios():
    marca = request.args.get("marca")
    modelo = request.args.get("modelo")
    precio_max = request.args.get("precio")
    anyo = request.args.get("anyo")
    tipo = request.args.get("tipo")

    query = ("SELECT id, marca, modelo, precio, anyo, tipo, autonomia, en_oferta, "
            "tiempo_carga, capacidad_bateria, tipo_hibrido, consumo_mixto, imagen_url FROM coches WHERE 1=1")
    params = []
```

```
conn = get_mariadb_connection()
with conn.cursor() as cursor:
    cursor.execute(query, tuple(params))
    rows = cursor.fetchall()
conn.close()

coches = [
    Coche(id=row[0], marca=row[1], modelo=row[2], precio=row[3], anyo=row[4], tipo=row[5].lower() if row[5]
        en_oferta=bool(row[7]), tiempo_carga=row[8], capacidad_bateria=row[9],
        tipo_hibrido=row[10], consumo_mixto=row[11], imagen_url=row[12])
    for row in rows
]
```

FAVORITS

Ara quan un usuari estigui buscant un cotxe té la possibilitat de guardar els que li interessin a la secció favorites. Això ho aconseguim fent ús de la funció **añadir_favorito**. D'aquesta manera l'id de l'usuari i del cotxe es guarda a la col·lecció de mongodb "favorites".

```

@app.route('/añadir_favorito', methods=['POST'])
@login_required
def añadir_favorito():
    anuncio_id = int(request.form.get('anuncio_id'))
    user_id = session.get("user_id")

    favorito = favoritos_col.find_one({'user_id': str(user_id)})

    if favorito:
        if anuncio_id not in favorito['coches']:
            favoritos_col.update_one(
                {'user_id': str(user_id)},
                {'$push': {'coches': anuncio_id}}
            )
    else:
        favoritos_col.insert_one({
            'user_id': str(user_id),
            'coches': [anuncio_id]
        })

    flash("Anuncio añadido a favoritos.")
    return redirect(url_for('favoritos'))

```

The screenshot shows a web application interface. On the left is a sidebar menu with the following items: 'Hife', 'TradeMotor' (expanded), 'comentarios', 'favoritos' (selected), 'historial_precios', 'ofertas', and 'admin'. The main content area displays two JSON objects representing favorite items:

```

{
  "_id": ObjectId('682c6587ad5fb1fb9adb2530'),
  "user_id": "6",
  "coches": Array (1)
    0: 37
}

```

```

{
  "_id": ObjectId('682cc3c92bb553f0d73bc78b'),
  "user_id": "1",
  "coches": Array (1)
}

```

OFERTA DEL COMPRADOR

Quan l'usuari vulgui realitzar una oferta pel cotxe podrà fer clic al boto "Hacer oferta", on serà enviat a un formulari on ha de posar el monto que vol pagar pel vehicle i a baix un comentari per al venedor.

Haz una oferta

Monto de la oferta (€):

Mensaje al vendedor:

Enviar oferta

Cancelar

Aquesta oferta serà enviada a la base de dades de mongodb i després serà enviada al venedor.

```
@app.route('/hacer_oferta/<int:anuncio_id>', methods=['GET', 'POST'])
@login_required
def hacer_oferta(anuncio_id):
    if request.method == 'POST':
        # Obtener el vendedor asociado a este anuncio desde la base de datos SQL
        conn = get_mariadb_connection()
        with conn.cursor() as cursor:
            cursor.execute("SELECT vendedor_id FROM coches WHERE id = %s", (anuncio_id,))
            vendedor = cursor.fetchone()
        conn.close()
        if not vendedor:
            flash("El anuncio no existe.", "error")
            return redirect(url_for('anuncios'))

        oferta = {
            'coche_id': anuncio_id,
            'user_id': session.get('user_id'),
            'nombre_usuario': session.get('nombre'),
            'monto': float(request.form['monto']),
            'mensaje': request.form['mensaje'],
            'estado': "pendiente", # Estado por defecto
            'vendedor_id': vendedor[0] # ID del vendedor obtenido desde MariaDB
        }

        ofertas_col.insert_one(oferta)
        flash('Oferta enviada correctamente!') # Mostrar mensaje (extra)
        return redirect(url_for('anuncios'))

    return render_template('hacer_oferta.html', anuncio_id=anuncio_id)
```

CONNECTIONS (2)

Search connections

- cluster0.q14yb.mongodb.net
 - localhost:27017
 - Hife
 - TradeMotor
 - comentarios
 - favoritos
 - historial_precios
 - ofertas**
 - admin
 - config
 - local
 - tasques_db
 - test

Type a query: { field: 'value' } or [Generate query](#)

ADD DATA **EXPORT DATA** **UPDATE** **DELETE**

```

_id: ObjectId('682c65a8ad5fb1fb9adb2531')
coche_id: 50
user_id: 6
nombre_usuario: "Rodrigo Salas"
monto: 41000
mensaje: "Buenos dias, ofrezco los 41000 por el coche."
estado: "aceptada"
vendedor_id: 1

```

```

_id: ObjectId('682cd181fe9dba085cc0e719')
coche_id: 37
user_id: 6
nombre_usuario: "Rodrigo Salas"
monto: 25979
mensaje: "Acepto el precio del anuncio."
estado: "aceptada"
vendedor_id: 1

```

VENEDOR ACCEPTA L'OFERTA

Ara accedirem al venedor per veure com es mostra la oferta, on aquest podrà acceptar la oferta o rebutjar-la. En aquest cas l'acceptem.

Editar **Eliminar**

Kia Niro
 Tipo: hibrido
 Precio: 25979.0 €
 Año: 2025
 Superoferta 21% de descuento

Ford Focus
 Tipo: gasoli
 Precio: 245
 Año: 2023

Ofertas de compradores:

Rodrigo Salas ofrece **25979.0 €**
 Acepto el precio del anuncio. **aceptada**

Rodrigo Salas ofrece **25900.0 €**
 Buenos dias, ofrezco 25900 por el!

Aceptar **Rechazar**

Ofertas de
 No hay ofer

I ens apareix com a oferta acceptada 👍. Ara l'oferta haurà pasat a transaccions, a mariadb

Editar

Eliminar

Kia Niro

Tipo: hibrido

Precio: 25979.0 €

Año: 2025

Superoferla 21% de descuento

Ofertas de compradores:

Rodrigo Salas ofrece 25979.0 €

Acepto el precio del anuncio.

aceptada

Rodrigo Salas ofrece 25900.0 €

Buenos días, ofrezco 25900 por el!

aceptada

Transacciones Enter a SQL expression to filter results (use Ctrl+Space)						
Grilla	id	comprador_id	coche_id	precio_final	fecha_venta	vendedor_id
1	5	6	50	41.000	2025-05-20 11:21:36.000	1
2	6	6	37	25.979	2025-05-20 19:04:42.000	1
3	7	6	37	25.900	2025-05-21 12:12:15.000	1
Texto						

RESSENYA DEL COTXE

Ara el comprador pot escriure una ressenya del cotxe, sol els usuaris que compren el cotxe poden realitzar comentaris per que els vegin els demés compradors. Per fer-ho he programat la funció **agregar comentario**.

```

@app.route('/agregar_comentario/<int:coche_id>', methods=['POST'])
@login_required
def agregar_comentario(coche_id):
    usuario_id = session.get("user_id")
    comentario = request.form.get("comentario", "").strip()
    if not comentario:
        flash("El comentario no puede estar vacío.", "error")
        return redirect(url_for("anuncios"))

    # Verificar si el usuario ha comprado este coche
    conn = get_mariadb_connection()
    with conn.cursor() as cursor:
        cursor.execute("SELECT COUNT(*) FROM transacciones WHERE comprador_id = %s AND coche_id = %s", (usuario_id, coche_id))
        ha_comprado = cursor.fetchone()[0] > 0
    conn.close()

    if not ha_comprado:
        flash("Solo los compradores pueden dejar comentarios.", "error")
        return redirect(url_for("anuncios"))

    nuevo_comentario = {
        "usuario_id": usuario_id,
        "comentario": comentario,
        "fecha": datetime.now().strftime("%Y-%m-%d")
    }

```

La ressenya s'escriu des de la pàgina "Mis compras" que comentarem més tard.

Mis Compras

Niro
Precio pagado: 25979.00 €
Fecha de compra: 2025-05-20 19:04:42

Escribe tu comentario aquí...

Enviar comentario

Kia Niro
Precio pagado: 25900.00 €
Fecha de compra: 2025-05-21 12:12:15

Me ha gustado este vehiculo!!

Enviar comentario

2025-05-21, 25900 €

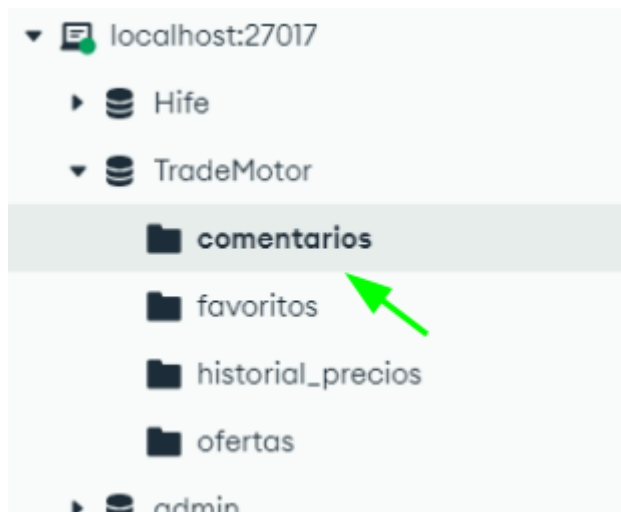
Comentarios:

Rodrigo Salas: Me ha gustado mucho este vehiculo, tiene buena autonomia y rendimiento por el precio pagado 🍷 . (2025-05-20 21:11:07.655000)

Rodrigo Salas: Me ha gustado este vehiculo!! (2025-05-21 14:30:53.197000)

Cerrar

Aquest comentari s'emmagatzema a la col·lecció comentarios, dins de la base de dades MongoDB.



```
_id: ObjectId('682dc77d00baad42a10579c8')
coche_id: 37
usuario_id: 6
comentario: "Me ha gustado este vehiculo!!"
fecha: 2025-05-21T14:30:53.197+00:00
```

HISTORIAL DE PREUS DEL COTXE

Per al historial de preus tenia dues idees, que s'actualitzi cada cop que es ven un cotxe amb el preu de la venta o que s'actualitzi quan el venedor vulgui modificar el preu del anunci, jo em vaig decantar per la segona idea, ja que si un comprador vol baixar el preu de la venta per motius diversos, com un tracte o altres raons, aquest preu reduït no es mostrarà als anuncis.

El venedor té la secció **vendes** on pot modificar el preu de l'anunci.

Editar BMW Serie 3

×

Tipo:

Diesel

▼

Precio (€):

49150,0

Año:

2023

Guardar cambios

Cancelar

Editar BMW Serie 3

×

Tipo:

Diesel

▼

Precio (€):

48500,0

Año:

2023

▲▼

Guardar cambios

Cancelar

A **anuncios** es veurà l'historial de preus que ha tingut aquest anunci.

BMW Serie 3

Tipo: diesel

Año: 2023

Precio: 48500.0 €

Historial de precios:

2025-05-21: 49150 €

2025-05-21: 48500 €

Comentarios:

Este anuncio aún no tiene comentarios.

Cerrar

Aquesta és la funció utilitzada:

```
historial_por_coche = {coche.id: obtener_historial_precios(coche.id) for coche in coches}

comentarios_por_anuncio = {}
for coche in coches:
    comentarios = list(comentarios_col.find({"coche_id": coche.id}))
    comentarios_con_nombres = []

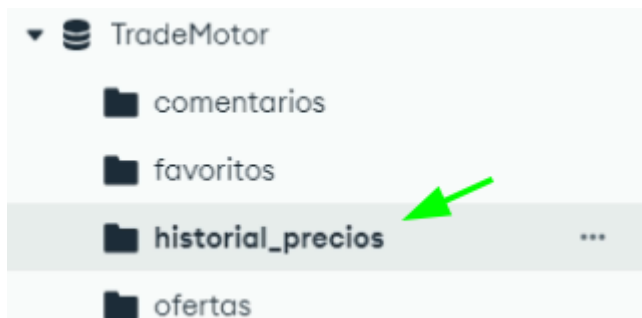
    conn = get_mariadb_connection()
    with conn.cursor() as cursor:
        for comentario in comentarios:
            cursor.execute("SELECT nombre FROM usuarios WHERE id = %s", (comentario.get("usuario_id"),))
            usuario = cursor.fetchone()
            nombre_usuario = usuario[0] if usuario else "Usuario desconocido"

            comentarios_con_nombres.append({
                "nombre_usuario": nombre_usuario,
                "texto": comentario.get("comentario", "Comentario no disponible"),
                "fecha": comentario.get("fecha", "Fecha desconocida")
            })
    conn.close()

    comentarios_por_anuncio[coche.id] = comentarios_con_nombres

return render_template("anuncios.html", anuncios=coches, comentarios_por_anuncio=comentarios_por_anuncio, historial_por_coche=historial_por_coche, active_page="anuncios")
```

Aquest historial es guardara a la col·lecció historial_precios dins de MongoDB.



```
_id: ObjectId('682dcb495099eb2b5b8a50c4')
coche_id: 48
▼ precios: Array (2)
  ▼ 0: Object
    fecha: "2025-05-21"
    precio: 49150
  ▼ 1: Object
    fecha: "2025-05-21"
    precio: 48500
```

EXTENSIONS DE L'APLICACIÓ

SECCIÓ DE VENDES

Com has pogut veure abans, s'han afegit apartats extra al programa, ara els venedors tenen una secció on poden gestionar les seves vendes, aquí podran editar els preus, eliminar els anuncis, veure les ofertes realitzades pels compradors...

Oferta aceptada y venta registrada correctamente.

Mis Coches en Venta

Editar

Eliminar

Tesla Model 3
Tipo: diesel
Precio: 53000.0 €
Año: 2025

Ofertas de compradores:

Rodrigo Salas ofrece **53000.0 €**
Buenos días, estoy interesado en comprar el Tesla Model 3. aceptada

Editar

Eliminar

Hyundai Ioniq 5 AWD
Tipo: electrico
Precio: 88800.0 €
Año: 2025

Ofertas de compradores:

Diego Hernandez ofrece **88800.0 €**
Buenos días, me gustaria ir a ver el vehiculo! Estoy interesado. Aceptar Rechazar

Editar

Eliminar

Toyota Corolla
Tipo: hibrido
Precio: 26500.0 €
Año: 2024
Superoferta 21% de descuento

Ofertas de compradores:
No hay ofertas aún.

Aquests venedors ara podran pujar imatges dels vehicles:

Publicar un nuevo anuncio

Tipo de vehículo:
--Selecciona una opción--

Marca:

Modelo:

Precio:

Año:

¿Está en oferta? ☐ Sí

Imágenes del coche:
Elegir archivos BMW-serie3.jpg

Publicar Anuncio

Les imatges es mostraren a uns nous models dels anuncis que s'accedeix fent clic sobre l'anunci:

Volkswagen ID4 Pro

Tipo: electrico


Año: 2025

Precio: 41000.0 €

Autonomía: 950.0 km

Tiempo de carga: 0.40 horas

Capacidad de batería: 74.00 kWh



Historial de precios:
No hay historial de precios disponible.

Comentarios:

Rodrigo Salas: Me gusta mucho este vehiculo, muy moderno y completo. (2025-05-20 13:22:20.206000)

Rodrigo Salas: El coche ha empezado a fallar!! (2025-05-20 19:12:07.850000)

Cerrar

MILLORES DE FILTRAT D'ANUNCIS

Ara la secció d'anuncis té un filtre més treballat on es poden filtrar tots els paràmetres, aquest s'adapta segons el tipus de vehicle:

32

Lista de anuncios

▼

Filtrar

Dacia Sandero Essential
Tipo: gasolina
Precio: 12540.0 €
Año: 2025

Superoferta 21% de descuento

Hacer Oferta

Ford Focus 1.0
Tipo: gasolina
Precio: 24500.0 €
Año: 2023

Hacer Oferta

Publicar un nuevo anuncio

Volver a la página principal

Lista de anuncios

▼

Filtrar

Volkswagen ID4 Pro
Tipo: electrico
Precio: 41000.0 €
Año: 2025

Hacer Oferta

Per últim els compradors tenen una secció on es guarden els cotxes que han comprat.

Mis Compras

Volkswagen ID4 Pro
Precio pagado: 41000.00 €
Fecha de compra: 2025-05-20 11:21:36

Escribe tu comentario aquí...

Enviar comentario

Kia Niro
Precio pagado: 25979.00 €
Fecha de compra: 2025-05-20 19:04:42

Escribe tu comentario aquí...

Enviar comentario

Kia Niro
Precio pagado: 25900.00 €
Fecha de compra: 2025-05-21 12:12:15

Escribe tu comentario aquí...

Enviar comentario

Tesla Model 3
Precio pagado: 53000.00 €
Fecha de compra: 2025-05-21 15:06:37

Escribe tu comentario aquí...

Enviar comentario

I ja que s'han afegit aquests apartats he incluit afegir un navbar que facilita la navegació pel programa.

TradeMotor

 Hola, Rodrigo Salas
 [Inicio](#)
[Ventas](#)
[Mis Compras](#)
[Favoritos](#)
[Cerrar sesión](#)

EXPLICACIÓ SOBRE LES MODIFICACIONS

Tot i que ha sigut un treball llarg, he pogut assolir totes les modificacions, fer millores i agregar extensions. He dedicat molt de temps en aquest treball i estic content sobre com ha quedat!, m'agradaria afegir més coses útils com un xat entre comprador i venedor, un sistema de pagament o seguretat però això ja formaria part d'una versió més avançada, l'important és que he après molt durant aquest temps!

