



# AEC3 - REDIS

🕒 Creado	@23 de noviembre de 2025 22:16
🏷️ Etiquetas	



## Descripción del Proyecto

Para esta actividad he desarrollado en Python un sistema de gestión para una flota de transportes (autobuses, furgonetas, vans) similar a empresas como Hertz o Enterprise, usando Redis como base de datos principal y `redis-py` para la implementación de la lógica.

### puntos a cubrir:

- Almacenar información de vehículos y reservas de forma eficiente.
- Gestionar sesiones de usuario y datos temporales (por ejemplo, reservas pendientes).
- Mantener contadores y métricas en tiempo real (vehículos disponibles, reservas activas, ingresos).
- Implementar rankings de vehículos más populares o clientes con mayor gasto.
- Gestionar colas de mantenimiento y notificaciones en tiempo real mediante Pub/Sub.

## Justificación de Redis en este escenario y su integración con python

Redis es especialmente adecuado para un sistema de gestión de flotas como este porque la inmediatez y la flexibilidad son esenciales: las operaciones en memoria permiten respuestas instantáneas para consultas de disponibilidad, reservas o notificaciones, algo crítico en escenarios de alta demanda y concurrencia. Además, la variedad de estructuras de datos de Redis (Strings, Hashes, Lists, Sets, Sorted Sets) facilita modelar entidades complejas como vehículos, reservas, sesiones y rankings de manera eficiente y natural.

En mi experiencia previa, ya había utilizado Redis junto con Python en otros proyectos, y siempre he encontrado que la integración entre ambos es muy fluida. Python, gracias a librerías como `redis-py`, permite explotar toda la potencia de Redis de forma sencilla y expresiva, acelerando el desarrollo y facilitando la escritura de lógica compleja sin sacrificar rendimiento. Esta combinación me ha permitido implementar desde sistemas de caché hasta colas de tareas y mecanismos de

notificación en tiempo real, beneficiándome tanto de la velocidad de Redis como de la versatilidad de Python.

## Estructura del Proyecto

```
AEC3/
├── README.md
├── AEC3_IHC.pdf
├── implementacion/
│   ├── 01_vehiculos.py      # Gestión de vehículos (Strings, Hash etc)
│   ├── 02_reservas.py      # Sistema de reservas (Hashes con TTL)
│   ├── 03_sesiones.py      # Gestión de sesiones (Strings con SETEX)
│   ├── 04_metricas.py      # Contadores y estadísticas (INCR, Hashes)
│   ├── 05_rankings.py      # Rankings dinámicos (Sorted Sets)
│   ├── 06_mantenimiento.py  # Colas de mantenimiento (Lists)
│   ├── 07_notificaciones.py # Sistema Pub/Sub en tiempo real
│   ├── 08_transacciones.py  # MULTI/EXEC, WATCH, Pipelining
│   └── 09_cache_policias.py # TTL, EXPIRE, Políticas de memoria
├── pruebas/
└── tester_isma_aec3.py      # Tester automatizado en Python
```

## Inicio Rápido

### Opción 1:

```
# 1. Iniciar Redisredis-server# 2. Instalar dependencias Pythonpip install redis
# 3. Ejecutar los módulos Python en ordenpython implementacion/01_vehiculos.py
python implementacion/02_reservas.py
python implementacion/03_sesiones.py
# ...etc
```

### Opción 2: Validación con Tester

```
python pruebas/tester_isma_aec3.py
```

## ¿Que he usado?

La solución implementa la mayoría de los comandos y estructuras estudiados en la unidad.

- Cadenas: SET, GET, INCR, APPEND para contadores y sesiones.
- Listas: LPUSH, RPUSH, LPOP, RPOP para colas de mantenimiento.
- Hashes: HSET, HGET, HDEL para vehículos y reservas.
- Sets y Sorted Sets: SADD, SMEMBERS, ZADD, ZRANGE para búsquedas y rankings.
- Pipelining y transacciones: MULTI, EXEC, WATCH para operaciones atómicas.
- Pub/Sub: canales de notificación en tiempo real.

- Caché y expiración: EXPIRE, TTL y configuración de políticas de memoria.

## Funcionalidades concretas

- **Pub/Sub:** Notificaciones en tiempo real
  - **Transacciones:** MULTI/EXEC para atomicidad
  - **Watch:** Optimistic locking para concurrencia
  - **TTL/Expiración:** Auto-limpieza de datos temporales
  - **Pipelining:** Optimización de latencia de red
  - **Políticas de Memoria:** allkeys-lru para gestión automática
- 

# Escenarios cubiertos y justificación

## Justificación de las estructuras y comandos

Para cada funcionalidad del sistema he elegido la estructura de datos de Redis que mejor se adapta al tipo de información y a la operación principal que se realiza:

- Los **vehículos** se almacenan en hashes, ya que permiten guardar todos los atributos relevantes bajo una sola clave y acceder o modificar cualquier campo de forma eficiente.
- Para la **búsqueda y filtrado** de vehículos por tipo, ciudad o estado, uso sets, que permiten agrupar y consultar rápidamente los identificadores según distintos criterios.
- Las **reservas** también se representan como hashes, y se les aplica un TTL para que expiren automáticamente si no se confirman, lo que es ideal para pre-reservas o bloqueos temporales.
- Las **sesiones de usuario** se gestionan con strings con expiración automática (SETEX), y los sets permiten asociar varias sesiones a un mismo usuario para soportar el acceso multi-dispositivo.
- Los **contadores** y métricas globales (vehículos disponibles, reservas activas, ingresos) se gestionan como strings para aprovechar las operaciones atómicas de incremento y decremento, mientras que las métricas agrupadas se almacenan en hashes.
- Los **rankings** (por popularidad, gasto, ingresos) se implementan con sorted sets, que permiten ordenar y consultar los elementos según su puntuación de forma eficiente.
- Para la **gestión de mantenimiento**, utilizo listas para las colas FIFO y sorted sets para priorizar vehículos según la urgencia.
- El sistema de **notificaciones** en tiempo real se basa en los canales Pub/Sub de Redis, que permiten enviar mensajes instantáneos a usuarios o grupos.
- Las **transacciones** y el **pipelining** se emplean para garantizar la atomicidad y eficiencia en operaciones que afectan a varios datos a la vez, como la confirmación de una reserva y la actualización de la disponibilidad.
- Finalmente, para la **caché** y los datos temporales, uso strings con TTL y configuro políticas de memoria LRU para asegurar que los datos más consultados permanezcan accesibles.

# Pruebas y validación

Con el tester que he creado podemos ejecutar de golpe los scripts de la carpeta `implementacion/` y se recogen los resultados principales de cada uno. A continuación se muestran los resultados obtenidos para cada archivo:

## 01\_vehiculos.py

```
=== REGISTRO DE VEHÍCULOS ===
V001: {'matricula': '1234ABC', 'marca': 'Mercedes', 'modelo': 'Sprinter', 'tipo': 'furgoneta', 'capacidad': '12', 'estado': 'disponible', 'ubicacion': 'Madrid Centro', 'km_actuales': '45000', 'año': '2022', 'precio_dia': '89.5', 'rating': '4.7'}
V002: {'matricula': '5678DEF', 'marca': 'Iveco', 'modelo': 'Daily', 'tipo': 'autobus', 'capacidad': '25', 'estado': 'disponible', 'ubicacion': 'Barcelona', 'km_actuales': '32000', 'año': '2023', 'precio_dia': '145.0', 'rating': '4.9'}
V003: {'matricula': '9012GHI', 'marca': 'Ford', 'modelo': 'Transit', 'tipo': 'van', 'capacidad': '8', 'estado': 'en_uso', 'ubicacion': 'Valencia', 'km_actuales': '28500', 'año': '2021', 'precio_dia': '75.0', 'rating': '4.5'}
V004: {'matricula': '3456JKL', 'marca': 'Renault', 'modelo': 'Master', 'tipo': 'furgoneta', 'capacidad': '14', 'estado': 'disponible', 'ubicacion': 'Sevilla', 'km_actuales': '52000', 'año': '2020', 'precio_dia': '82.0', 'rating': '4.3'}
V005: {'matricula': '7890MNO', 'marca': 'MAN', 'modelo': 'Lion's City', 'tipo': 'autobus', 'capacidad': '40', 'estado': 'mantenimiento', 'ubicacion': 'Madrid Taller', 'km_actuales': '85000', 'año': '2019', 'precio_dia': '180.0', 'rating': '4.6'}

=== CONSULTAS ===
Info V001: {'matricula': '1234ABC', 'marca': 'Mercedes', 'modelo': 'Sprinter', 'tipo': 'furgoneta', 'capacidad': '12', 'estado': 'disponible', 'ubicacion': 'Madrid Centro', 'km_actuales': '45000', 'año': '2022', 'precio_dia': '89.5', 'rating': '4.7'}
Campos V001 (marca, modelo, precio_dia, estado, ubicacion): ['Mercedes', 'Sprinter', '89.5', 'disponible', 'Madrid Centro']
Estado V001: disponible
¿Existe matrícula en V001?: True

=== ACTUALIZACIONES ===
V001 tras actualización: {'matricula': '1234ABC', 'marca': 'Mercedes', 'modelo': 'Sprinter', 'tipo': 'furgoneta', 'capacidad': '12', 'estado': 'mantenimiento', 'ubicacion': 'Madrid Taller', 'km_actuales': '45000', 'año': '2022', 'precio_dia': '89.5', 'rating': '4.8'}
V003 tras actualización: {'matricula': '9012GHI', 'marca': 'Ford', 'modelo': 'Transit', 'tipo': 'van', 'capacidad': '8', 'estado': 'disponible', 'ubicacion': 'Valencia Centro', 'km_actuales': '28750', 'año': '2021', 'precio_dia': '75.0', 'rating': '4.5'}

=== CONTADORES ===
Total vehículos: 5
Vehículos disponibles: 3
En uso: 1
En mantenimiento: 1
```

=== SETS POR TIPO ===

Furgonetas: {'V001', 'V004'}

¿V001 es furgoneta?: 1

Nº furgonetas: 2

=== DISPONIBILIDAD POR CIUDAD Y ESTADO ===

Disponibles en Madrid: {'V001'}

Furgonetas disponibles: {'V001', 'V004'}

=== CACHÉ VEHÍCULO V001 ===

Cache V001: {"id":"V001","marca":"Mercedes","modelo":"Sprinter","precio\_dia":89.50,"disponible":true}

TTL cache V001: 300

=== OPERACIONES ADICIONALES SOBRE V001 ===

Campos: ['matricula', 'marca', 'modelo', 'tipo', 'capacidad', 'estado', 'ubicacion', 'km\_actuales', 'año', 'precio\_dia', 'rating']

Valores: ['1234ABC', 'Mercedes', 'Sprinter', 'furgoneta', '12', 'mantenimiento', 'Madrid Taller', '45150', '2022', '95', '4.8']

Nº campos: 11

V001 tras incrementos: {'matricula': '1234ABC', 'marca': 'Mercedes', 'modelo': 'Sprinter', 'tipo': 'furgoneta', 'capacidad': '12', 'estado': 'mantenimiento', 'ubicacion': 'Madrid Taller', 'km\_actuales': '45150', 'año': '2022', 'precio\_dia': '95', 'rating': '4.8'}

=== DETALLES EXTENDIDOS V001 ===

{'color': 'blanco', 'matriculacion': '2022-03-15', 'seguro': 'Seguro123', 'revision\_proxima': '2025-12-01', 'propietario': 'FleetHub Madrid', 'contacto\_taller': '+34600123456'}

=== LISTADO GENERAL DE VEHÍCULOS ===

Claves: ['vehiculo:V002:reservas', 'vehiculo:V001', 'vehiculo:V002', 'vehiculo:V003:reservas', 'vehiculo:V005', 'vehiculo:V001:reservas', 'vehiculo:V001:detalles', 'vehiculo:V003', 'vehiculo:V004']

V001: ['1234ABC', 'Mercedes', 'Sprinter', 'mantenimiento', 'Madrid Taller']

V002: ['5678DEF', 'Iveco', 'Daily', 'disponible', 'Barcelona']

V003: ['9012GHI', 'Ford', 'Transit', 'disponible', 'Valencia Centro']

## 02\_reservas.py

=== CREACIÓN DE RESERVAS ===

Reserva R001: {'vehiculo\_id': 'V001', 'cliente\_id': 'C123', 'cliente\_nombre': 'Juan García', 'cliente\_email': 'juan@email.com', 'fecha\_inicio': '2025-11-20', 'fecha\_fin': '2025-11-23', 'dias': '3', 'precio\_total': '268.5', 'estado': 'confirmada', 'ubicacion\_recogida': 'Madrid Centro', 'ubicacion\_entrega': 'Madrid Centro', 'tipo\_reserva': 'mudanza', 'hora\_recogida': '2025-11-20T09:30:00', 'hora\_devolucion': '2025-11-23T18:45:00', 'km\_recorridos': '450', 'extension': 'true'}

Reserva R002: {'vehiculo\_id': 'V002', 'cliente\_id': 'C456', 'cliente\_nombre': 'María López', 'cliente\_email': 'maria@email.com', 'fecha\_inicio': '2025-11-21', 'fecha\_fin': '2025-11-21', 'dias': '1', 'precio\_total': '145.0', 'estado': 'confirmada', 'ubicacion\_recogida': 'Barcelona', 'ubicacion\_entrega': 'Barcelona'}

a': 'Barcelona', 'tipo\_reserva': 'excursion', 'motivo\_cancelacion': 'Cliente enfermo', 'fecha\_cancelacion': '2025-11-20T16:00:00'}

Reserva R003: {'vehiculo\_id': 'V003', 'cliente\_id': 'C789', 'cliente\_nombre': 'Tech Solutions SL', 'cliente\_email': 'info@techsol.com', 'fecha\_inicio': '2025-11-20', 'fecha\_fin': '2025-11-25', 'dias': '5', 'precio\_total': '375.0', 'estado': 'activa', 'ubicacion\_recogida': 'Valencia', 'ubicacion\_entrega': 'Valencia', 'tipo\_reserva': 'corporativa', 'ubicacion\_actual': 'Albacete', 'ultima\_actualizacion': '2025-11-21T14:20:00'}

#### === CONSULTAS ===

Detalles R001: {'vehiculo\_id': 'V001', 'cliente\_id': 'C123', 'cliente\_nombre': 'Juan García', 'cliente\_email': 'juan@email.com', 'fecha\_inicio': '2025-11-20', 'fecha\_fin': '2025-11-23', 'dias': '3', 'precio\_total': '268.5', 'estado': 'confirmada', 'ubicacion\_recogida': 'Madrid Centro', 'ubicacion\_entrega': 'Madrid Centro', 'tipo\_reserva': 'mudanza', 'hora\_recogida': '2025-11-20T09:30:00', 'hora\_devolucion': '2025-11-23T18:45:00', 'km\_recorridos': '450', 'extension': 'true'}

Info básica R001: ['Juan García', 'V001', '2025-11-20', '2025-11-23', 'confirmada']

Estado R001: confirmada

TTL R001: 259200

#### === ACTUALIZACIONES ===

R001 tras actualización: {'vehiculo\_id': 'V001', 'cliente\_id': 'C123', 'cliente\_nombre': 'Juan García', 'cliente\_email': 'juan@email.com', 'fecha\_inicio': '2025-11-20', 'fecha\_fin': '2025-11-23', 'dias': '3', 'precio\_total': '268.5', 'estado': 'completada', 'ubicacion\_recogida': 'Madrid Centro', 'ubicacion\_entrega': 'Madrid Centro', 'tipo\_reserva': 'mudanza', 'hora\_recogida': '2025-11-20T09:30:00', 'hora\_devolucion': '2025-11-23T18:45:00', 'km\_recorridos': '450', 'extension': 'true'}

R002 tras actualización: {'vehiculo\_id': 'V002', 'cliente\_id': 'C456', 'cliente\_nombre': 'María López', 'cliente\_email': 'maria@email.com', 'fecha\_inicio': '2025-11-21', 'fecha\_fin': '2025-11-21', 'dias': '1', 'precio\_total': '145.0', 'estado': 'cancelada', 'ubicacion\_recogida': 'Barcelona', 'ubicacion\_entrega': 'Barcelona', 'tipo\_reserva': 'excursion', 'motivo\_cancelacion': 'Cliente enfermo', 'fecha\_cancelacion': '2025-11-20T16:00:00'}

R003 tras actualización: {'vehiculo\_id': 'V003', 'cliente\_id': 'C789', 'cliente\_nombre': 'Tech Solutions SL', 'cliente\_email': 'info@techsol.com', 'fecha\_inicio': '2025-11-20', 'fecha\_fin': '2025-11-25', 'dias': '5', 'precio\_total': '375.0', 'estado': 'activa', 'ubicacion\_recogida': 'Valencia', 'ubicacion\_entrega': 'Valencia', 'tipo\_reserva': 'corporativa', 'ubicacion\_actual': 'Albacete', 'ultima\_actualizacion': '2025-11-21T14:20:00'}

#### === ÍNDICE POR CLIENTE ===

Reservas de C123 antes de borrar: {'R200', 'R001'}

Reservas de C123 después de borrar: {'R200'}

#### === ÍNDICE POR VEHÍCULO ===

Reservas de V001: {'R200', 'R001'}

Nº reservas V001: 2

#### === CONTADORES ===

Total: 4

Activas: 2

Completadas: 1

Canceladas: 1

Reservas día 2025-11-20: 5  
Reservas mes 2025-11: 16

=== COLA DE RESERVAS PENDIENTES ===

Pendientes: ['R006', 'R005', 'R004', 'R006', 'R005', 'R004']  
Nº pendientes: 6

=== EXTENSIÓN DE RESERVA ===

R001 tras extensión: {'vehiculo\_id': 'V001', 'cliente\_id': 'C123', 'cliente\_nombre': 'Juan García', 'cliente\_email': 'juan@email.com', 'fecha\_inicio': '2025-11-20', 'fecha\_fin': '2025-11-25', 'dias': '5', 'precio\_total': '447.5', 'estado': 'completada', 'ubicacion\_recogida': 'Madrid Centro', 'ubicacion\_entrega': 'Madrid Centro', 'tipo\_reserva': 'mudanza', 'hora\_recogida': '2025-11-20T09:30:00', 'hora\_devolucion': '2025-11-23T18:45:00', 'km\_recorridos': '450', 'extension': 'true'}  
Nuevo TTL R001: 432000

=== PRE-RESERVAS ===

Pre-reserva PR001: {'vehiculo\_id': 'V001', 'cliente\_id': 'C999', 'fecha\_inicio': '2025-11-25', 'fecha\_fin': '2025-11-27', 'precio\_estimado': '179.0', 'estado': 'pendiente\_pago'}  
Reserva R010: {'vehiculo\_id': 'V001', 'cliente\_id': 'C999', 'fecha\_inicio': '2025-11-25', 'fecha\_fin': '2025-11-27', 'dias': '2', 'precio\_total': '179.0', 'estado': 'confirmada'}

=== HISTORIAL DE RESERVAS ===

Historial de C123: {'R001', 'R015', 'R007'}  
Historial R001: {'vehiculo\_id': 'V001', 'cliente\_id': 'C123', 'fecha\_inicio': '2025-11-20', 'fecha\_fin': '2025-11-23', 'precio\_total': '268.5', 'estado': 'completada', 'rating\_cliente': '5', 'comentario': 'Excelente servicio'}

### 03\_sesiones.py

=== CREACIÓN DE SESIONES (STRINGS) ===

Session user: {"user\_id": "C123", "tipo": "cliente", "nombre": "Juan García", "email": "juan@email.com", "login": "2025-11-20T10:00:00", "ultima\_actividad": "2025-11-20T10:15:00", "permisos": ["ver\_vehiculos", "hacer\_reservas", "ver\_historial"]}  
Session driver: {"user\_id": "D456", "tipo": "conductor", "nombre": "Carlos Ruiz", "vehiculo\_asignado": "V001", "estado": "activo", "login": "2025-11-20T08:00:00", "ubicacion": "Madrid Centro"}  
Session admin: {"user\_id": "A789", "tipo": "admin", "nombre": "Laura Martín", "nivel": "super\_admin", "login": "2025-11-20T09:00:00", "ip": "192.168.1.100"}

=== VERIFICACIÓN Y TTL ===

Session data: {"user\_id": "C123", "tipo": "cliente", "nombre": "Juan García", "email": "juan@email.com", "login": "2025-11-20T10:00:00", "ultima\_actividad": "2025-11-20T10:15:00", "permisos": ["ver\_vehiculos", "hacer\_reservas", "ver\_historial"]}  
TTL session user: 1800

=== LOGOUT / CIERRE DE SESIÓN ===

¿Existe session:user:abc123def456?: 0

=== ÍNDICE DE SESIONES POR USUARIO ===

Sesiones activas de C123: {'session:user:xyz789', 'session:user:mobile789xyz', 'session:user:abc123def456'}

=== CONTADOR DE SESIONES ACTIVAS ===

Cientes activas: 0

Conductores activas: 0

Admins activas: 0

=== SESIONES DETALLADAS (HASHES) ===

Campos clave: ['C123', 'cliente', '2025-11-20T10:45:00']

Todos los campos: {'user\_id': 'C123', 'tipo': 'cliente', 'nombre': 'Juan García', 'email': 'juan@email.com', 'login': '2025-11-20T10:00:00', 'ultima\_actividad': '2025-11-20T10:45:00', 'ip': '192.168.1.50', 'dispositivo': 'Chrome/Windows', 'token\_version': '1'}

=== BLOQUEO DE CUENTAS ===

¿C123 bloqueada temporal?: true

¿C999 bloqueada permanente?: permanent

=== INTENTOS DE LOGIN Y BLOQUEO ===

Intentos de login juan@email.com: 1

¿Login bloqueado?: true

=== TOKEN DE RECUPERACIÓN ===

Reset token: {"email":"juan@email.com","user\_id":"C123","created":"2025-11-20T11:00:00"}

=== SESIÓN DE CONDUCTOR ===

Driver session D456: {'driver\_id': 'D456', 'nombre': 'Carlos Ruiz', 'vehiculo': 'V001', 'estado': 'en\_pausa', 'reserva\_actual': 'R001', 'ubicacion': 'Madrid Centro', 'inicio\_turno': '2025-11-20T08:00:00', 'ultima\_ubicacion\_update': '2025-11-20T12:31:00'}

## 04\_mtricas.py

=== CONTADORES GLOBALES ===

Total vehículos: 50

Vehículos disponibles: 38

Reservas activas: 10

Cientes registrados: 892

=== OPERACIONES SOBRE CONTADORES ===

Vehículos disponibles: 38

Vehículos en uso: 10

Vehículos en mantenimiento: 2

Reservas activas: 10

=== CONTADORES DE KILÓMETROS ===

Km hoy: 3570



Km mes actual: 45800  
Km total histórico: 2500120

=== CONTADORES DE INGRESOS ===

Ingresos día: 310  
Ingresos mes: 45768.3999999999999999858  
Ingresos año: 456878.9499999999999998863

=== MÉTRICAS POR TIPO DE VEHÍCULO ===

Furgoneta: {'total': '25', 'disponibles': '17', 'en\_uso': '6', 'mantenimiento': '2', 'reservas\_mes': '146', 'ingresos\_mes': '12980'}  
Disponibles furgoneta: 17

=== MÉTRICAS POR CIUDAD ===

Madrid: {'vehiculos': '20', 'disponibles': '14', 'en\_uso': '5', 'reservas\_hoy': '9', 'ingresos\_hoy': '768'}  
Barcelona: {'vehiculos': '15', 'disponibles': '10', 'en\_uso': '4', 'reservas\_hoy': '6', 'ingresos\_hoy': '523.0'}

=== RANKING VEHÍCULOS RENTABLES ===

Top 5 rentables: [('V005', 6123.9), ('V003', 5234.75), ('V001', 4567.5), ('V002', 3892.0), ('V004', 2890.3)]  
Posición V003: 1  
>4000 ingresos: [('V001', 4657.0), ('V003', 5234.75), ('V005', 6123.9)]

=== RANKING VEHÍCULOS POPULARES ===

Top 10 populares: [('V003', 52.0), ('V001', 45.0), ('V005', 41.0), ('V002', 38.0), ('V004', 31.0)]  
Menos populares: [('V004', 31.0), ('V002', 38.0), ('V005', 41.0), ('V001', 46.0), ('V003', 52.0)]

=== RANKING CLIENTES POR GASTO ===

Top 10 clientes: [('C012', 3456.9), ('C456', 2567.8), ('C123', 1234.5), ('C789', 892.3), ('C345', 567.25)]  
VIPs (>2000): [('C456', 2567.8), ('C012', 3456.9)]  
Clientes >1000: 3

=== RENDIMIENTO DE CONDUCTORES ===

D456: {'nombre': 'Carlos Ruiz', 'servicios\_completados': '235', 'km\_recorridos': '12705', 'rating\_promedio': '4.8', 'incidencias': '2', 'horas\_trabajadas': '859.5'}

## 05\_rankings.py

=== RANKING POPULARIDAD (RESERVAS) ===

Top 5 reservados: [('V007', 201.0), ('V003', 189.0), ('V010', 178.0), ('V005', 167.0), ('V001', 156.0)]  
Posición V004: 9  
Score V004 tras incremento: 99.0

=== RANKING INGRESOS ===

Top 3 ingresos: [('V003', 28934.25), ('V005', 24567.3), ('V001', 23456.8)]

>=20k y <=30k: [('V001', 23456.8), ('V005', 24567.3), ('V003', 28934.25)]

>=25k: [('V003', 28934.25)]

Nº >=20k: 3

=== RANKING RATING ===

Top 5 rating: [('V006', 4.95), ('V002', 4.9), ('V008', 4.85), ('V001', 4.8), ('V005', 4.7)]

Peor valorados: [('V004', 4.3), ('V007', 4.4), ('V003', 4.6)]

Rating >4.5: [('V003', 4.6), ('V005', 4.7), ('V001', 4.8), ('V008', 4.85), ('V002', 4.9), ('V006', 4.95)]

=== RANKING DISPONIBILIDAD ===

Más disponibles: [('V003', 30.0), ('V006', 29.0), ('V001', 28.0), ('V005', 27.0), ('V002', 25.0)]

Menos disponibles: [('V004', 22.0), ('V007', 24.0), ('V002', 25.0)]

=== RANKING CLIENTES FIELES ===

Top 10 fieles: [('C678', 89.0), ('C012', 67.0), ('C456', 45.0), ('C345', 34.0), ('C123', 23.0), ('C901', 19.0), ('C789', 12.0)]

VIPs (>50 reservas): [('C012', 67.0), ('C678', 89.0)]

Nuevos clientes (<=5): []

=== RANKING CLIENTES POR GASTO ===

Top 10 gasto: [('C012', 8901.2), ('C456', 5678.5), ('C345', 3456.8), ('C123', 2456.9), ('C789', 1234.75)]

Gold (>5000): [('C456', 5678.5), ('C012', 8901.2)]

Platinum (>10000): []

=== RANKING CONDUCTORES EFICIENCIA ===

Más eficientes: [('D012', 13.2), ('D678', 12.9), ('D456', 12.5), ('D789', 11.8), ('D345', 10.9)]

Menos eficientes: [('D345', 10.9), ('D789', 11.8), ('D456', 12.5)]

=== RANKING CONDUCTORES VALORACIÓN ===

Mejores conductores: [('D012', 4.95), ('D456', 4.9), ('D678', 4.85), ('D789', 4.7), ('D345', 4.6)]

Bajo 4.7: [('D345', 4.6), ('D789', 4.7)]

=== RANKING HOY (TEMPORAL) ===

Populares hoy: [('V005', 6.0), ('V002', 5.0), ('V004', 4.0), ('V001', 3.0), ('V003', 2.0)]

=== RANKING SOPORTE (TIEMPO RESPUESTA) ===

Agentes rápidos: [('agente002', 38.0), ('agente004', 41.0), ('agente001', 45.0), ('agente003', 52.0), ('agente005', 67.0)]

Agentes lentos: [('agente005', 67.0), ('agente003', 52.0), ('agente001', 45.0)]

## 06\_mantenimiento.py

Cola actual: ['V001', 'V005', 'V025', 'V005', 'V001', 'V005', 'V007', 'V012']  
 Pendientes: 8  
 Siguiente: V001  
 Próximos 3: ['V005', 'V025', 'V005']

Atendido urgente: V019  
Cola urgente actual: ['V023', 'V012', 'V007', 'V023', 'V012', 'V007', 'V023']

Revisiones: ['V008', 'V015', 'V022', 'V008', 'V015', 'V022']  
 Reparaciones: ['V017', 'V011', 'V017']  
 Limpieza: ['V020', 'V003', 'V009', 'V014', 'V020', 'V003', 'V009', 'V014', 'V020']  
 Neumáticos: ['V006', 'V018', 'V006', 'V018', 'V006', 'V018']

V001: {'vehiculo\_id': 'V001', 'tipo': 'revision\_programada', 'fecha\_entrada': '2025-11-20T09:00:00', 'km\_actuales': '45150', 'descripcion': 'Revisión 45.000 km', 'prioridad': 'normal', 'mecanico\_asignado': 'Juan Pérez', 'estado': 'en\_proceso', 'coste\_estimado': '350.0'}

V023: {'vehiculo\_id': 'V023', 'tipo': 'averia', 'fecha\_entrada': '2025-11-20T14:30:00', 'km\_actuales': '67890', 'descripcion': 'Motor hace ruido extraño', 'prioridad': 'urgente', 'mecanico\_asignado': 'Laura Gómez', 'estado': 'diagnostico', 'coste\_estimado': '0'}

V008: {'vehiculo\_id': 'V008', 'tipo': 'ITV', 'fecha\_entrada': '2025-11-20T08:00:00', 'km\_actuales': '32000', 'descripcion': 'ITV anual', 'prioridad': 'normal', 'mecanico\_asignado': 'Carlos Ruiz', 'estado': 'pendiente', 'coste\_estimado': '45.0'}

Últimos 5: [{"fecha":"2025-07-10","tipo":"revision","km":40000,"coste":320.00}', {"fecha":"2025-09-15","tipo":"reparacion","km":42300,"coste":567.80}', {"fecha":"2025-11-20","tipo":"revision","km":45150,"coste":385.50}', {"fecha":"2025-11-20","tipo":"revision","coste":385.50}', {"fecha":"2025-07-10","tipo":"revision","km":40000,"coste":320.00}']

Más reciente: {"fecha":"2025-07-10","tipo":"revision","km":40000,"coste":320.00}

Total historial: 11

```
V001 en mantenimiento: 1
V002 en mantenimiento: 0
Todos en mantenimiento: {'V001', 'V008', 'V023'}
Cuántos en mantenimiento: 3
```

Días por vehículo: {'V001': '3', 'V008': '1', 'V023': '5', 'V011': '2', 'V017': '4'}

Costes por vehículo: {'V001': '1636', 'V008': '890.3', 'V023': '2567.8'}

Coste mes 2025-11: 953.2999999999999

=== ALERTAS DE MANTENIMIENTO PREVENTIVO ===

>50.000 km: [('V012', 51000.0)]

Próximos a revisión (45-47k): [('V001', 46000.0)]

=== PRIORIZACIÓN DE MANTENIMIENTO ===

Más urgente: ['V023']

Todos priorizados: [('V011', 8.0), ('V008', 5.0), ('V001', 3.0), ('V017', 2.0)]

## 07\_notificaciones.py

=== NOTIFICACIONES A CONDUCTORES ===

Mensajes enviados a canal drivers.

=== NOTIFICACIONES A ADMINISTRADORES ===

Mensajes enviados a canal admins.

=== NOTIFICACIONES A CLIENTES ===

Mensajes enviados a cliente C123.

=== NOTIFICACIONES POR EVENTO ===

Mensajes enviados a eventos de reservas.

=== NOTIFICACIONES DE TALLER ===

Mensajes enviados a canal taller.

=== ALERTAS DE VEHÍCULOS EN TIEMPO REAL ===

Mensajes enviados a tracking:vehicles.

=== NOTIFICACIONES POR PATRÓN (WILDCARD) ===

Mensajes enviados a clientes C123 y C456.

=== NOTIFICACIONES POR CIUDAD ===

Mensajes enviados a Madrid y Barcelona.

=== BROADCAST GENERAL ===

Mensajes enviados a todos los usuarios.

=== NOTIFICACIONES DE EMERGENCIA ===

Mensajes enviados a canal de alertas de emergencia.

## 08\_transacciones.py

=== TRANSACCIÓN: CREAR RESERVA ===

Reserva R100: {'vehiculo\_id': 'V001', 'cliente\_id': 'C123', 'precio': '89.5', 'estado': 'confirmada'}

=== TRANSACCIÓN CANCELADA (DISCARD) ===

Reserva R101 existe: 0

=== TRANSACCIÓN: COMPLETAR RESERVA ===

Reserva R001: {'vehiculo\_id': 'V001', 'cliente\_id': 'C123', 'cliente\_nombre': 'Juan García', 'cliente\_email': 'juan@email.com', 'fecha\_inicio': '2025-11-20', 'fecha\_fin': '2025-11-20T18:00:00', 'dias': '5', 'precio\_total': '447.5', 'estado': 'completada', 'ubicacion\_recogida': 'Madrid Centro', 'ubicacion\_entrega': 'Madrid Centro', 'tipo\_reserva': 'mudanza', 'hora\_recogida': '2025-11-20T09:30:00', 'hora\_devolucion': '2025-11-23T18:45:00', 'km\_recorridos': '450', 'extension': 'true'}

=== TRANSACCIÓN: VEHÍCULO A MANTENIMIENTO ===

Vehículo V005: {'matricula': '7890MNO', 'marca': 'MAN', 'modelo': "Lion's City", 'tipo': 'autobus', 'capacidad': '40', 'estado': 'mantenimiento', 'ubicacion': 'Madrid Taller', 'km\_actuales': '85000', 'año': '2019', 'precio\_dia': '180.0', 'rating': '4.6'}

=== TRANSACCIÓN CON WATCH ===

Reserva R102: {'vehiculo\_id': 'V001', 'cliente\_id': 'C123', 'estado': 'confirmada'}

=== TRANSACCIÓN: ACTUALIZAR MÉTRICAS ===

Reservas hoy: 6

Reservas mes: 17

Total histórico: 1548

=== TRANSACCIÓN: CAMBIO DE ESTADO COMPLEJO ===

Vehículo V005: {'matricula': '7890MNO', 'marca': 'MAN', 'modelo': "Lion's City", 'tipo': 'autobus', 'capacidad': '40', 'estado': 'disponible', 'ubicacion': 'Madrid Centro', 'km\_actuales': '85000', 'año': '2019', 'precio\_dia': '180.0', 'rating': '4.6'}

=== PIPELINING ===

key1: value1

key2: value2

key3: value3

counter: 3

=== PIPELINE + TRANSACCIÓN ===

Reserva R103: {'vehiculo\_id': 'V002', 'cliente\_id': 'C789', 'precio': '145.0'}

=== TRANSACCIÓN: TRANSFERENCIA DE PUNTOS ===

Puntos C123: -150

Puntos C456: 150

Última transacción: {"de": "C123", "a": "C456", "cantidad": 50, "fecha": "2025-11-20"}

## 09\_cache\_policies.py

=== ESTRATEGIA DE TTL Y CACHÉ ===

TTL session:user:abc123: 1800

Cache vehículo V001: {"marca": "Mercedes", "modelo": "Sprinter"}

=== VERIFICACIÓN Y GESTIÓN DE TTL ===

TTL session:user:abc123: 1800

PTTL session:user:abc123: 1799999

TTL vehiculo:V001: -1

=== EXPIRACIÓN CON SETEX/PSSETEX ===

TTL cache:key: 300

=== PATRÓN CACHE-ASIDE ===

Cache vehículo V001: {"marca":"Mercedes","modelo":"Sprinter"}

=== INVALIDACIÓN DE CACHE ===

Cache vehículo V001 tras delete: None

=== CACHE CON ACTUALIZACIÓN PROACTIVA ===

Cache vehículo V001 actualizado: {"marca":"Mercedes","modelo":"Sprinter","precio\_dia":95.00}

=== EXPIRACIÓN POR LOTES ===

session:user:abc123 existe: 0

session:user:def456 existe: 0

session:user:ghi789 existe: 0

## Mi conclusión

Creo que la combinación de Redis y Python ha demostrado ser una solución muy potente para la gestión de flotas, permitiendo implementar funcionalidades avanzadas con gran rapidez y eficiencia. Esta arquitectura es fácilmente escalable y adaptable a nuevos requisitos, lo que la convierte en una base sólida para sistemas de gestión modernos.