



EJERCICIO 1 [1 punto]

```
def lee_gastos(fichero):
    res = []
    with open(fichero, encoding='utf-8') as f:
        lector = csv.reader(f)
        next(lector)
        for num_gasto, pagador, concepto, destinatario, importe, fecha in lector:
            num_gasto = int(num_gasto)
            importe = float(importe)
            fecha = datetime.strptime(fecha, '%d/%m/%Y').date()
            res.append(Gasto(num_gasto, pagador, concepto,
                             destinatario, importe, fecha))
    return res
```

EJERCICIO 2 [1 punto]

```
def pagadores_y_conceptos(gastos):
    return sorted({g.pagador for g in gastos}), sorted({g.concepto for g in gastos})
```

EJERCICIO 3 [1,5 puntos]

```
def total_importe(gastos, fecha_inicial=None, fecha_final=None):
    if fecha_inicial:
        fecha_inicial = datetime.strptime(fecha_inicial, '%d/%m/%Y').date()
    if fecha_final:
        fecha_final = datetime.strptime(fecha_final, '%d/%m/%Y').date()
    return sum(g.importe for g in gastos
               if (fecha_inicial == None or fecha_inicial <= g.fecha)
                   and (fecha_final == None or g.fecha <= fecha_final))
```

EJERCICIO 4 [1,5 puntos]

```
def conceptos_menos_gastos(gastos):
    total_por_gastos = Counter(g.concepto for g in gastos)
    min_gastos = min(total_por_gastos.values())
    return [concepto for concepto, freq in total_por_gastos.items()
            if freq == min_gastos]
```

EJERCICIO 5 [2 puntos]

```
def pagadores_mayor_importe_medio(gastos, n):
    totales_por_usuario = acumula_importes_por_usuario(gastos)
    contador_por_usuario = Counter(g.pagador for g in gastos)
    dicc_medias = {}
    for u in totales_por_usuario:
        dicc_medias[u] = totales_por_usuario[u] / contador_por_usuario[u]
    return sorted(dicc_medias.items(), reverse=True, key = lambda t:t[1])[:n]
```



#Versión alternativa por compresión

```
def pagadores_mayor_importe_medio(gastos, n):
    totales_por_usuario = acumula_importes_por_usuario(gastos)
    contador_por_usuario = Counter(g.pagador for g in gastos)
    dicc_medias = {u: totales_por_usuario[u] / contador_por_usuario[u]
                   for u in totales_por_usuario}
    return sorted(dicc_medias.items(), reverse=True, key = lambda t:t[1])[:n]
```

#función auxiliar para creación de diccionario

```
def acumula_importes_por_usuario(gastos):
    totales_por_usuario = {}
    for g in gastos:
        if g.pagador in totales_por_usuario:
            totales_por_usuario[g.pagador] += g.importe
        else:
            totales_por_usuario[g.pagador] = g.importe
    return totales_por_usuario
```

EJERCICIO 6 [2 puntos]

```
def balance(gastos):
    usuarios = {g.pagador for g in gastos}
    balance_por_usuario = {u:0 for u in usuarios}
    for g in gastos:
        if g.destinatario == 'Todos':
            aux_actualiza_balance(balance_por_usuario, g.importe, g.pagador, usuarios)
        else:
            aux_actualiza_balance(balance_por_usuario, g.importe,
                                  g.pagador, {g.pagador, g.destinatario})
    for usuario, balance in balance_por_usuario.items():
        balance_por_usuario[usuario] = round(balance, 2)
    return balance_por_usuario
```

#Función auxiliar

```
def aux_actualiza_balance(balance_por_usuario, importe, usuario, beneficiarios):
    balance_por_usuario[usuario] += importe
    importe_por_persona = importe / len(beneficiarios)
    for u in beneficiarios:
        balance_por_usuario[u] -= importe_por_persona
```

EJERCICIO 7 [1 punto]

```
def test_lee_gastos(gastos):
    print("Número de registros leídos:", len(gastos))
    print("Primer registro:", gastos[0])
```



```
def test_pagadores_y_conceptos(gastos):
    pagadores, conceptos = pagadores_y_conceptos(gastos)
    print("Pagadores:", pagadores)
    print("Conceptos:", conceptos)

def test_total_importe(gastos):
    print("La cantidad total gastada entre el 5 y el 8 de abril de 2019 fue:",
          total_importe(gastos, "05/04/2019", "08/04/2019"))
    print("La cantidad total gastada fue:", total_importe(gastos))

def test_conceptos_menos_gastos(gastos):
    print("Los conceptos con menos gastos registrados son:",
          conceptos_menos_gastos(gastos))

def test_pagadores_mayor_importe_medio(gastos):
    res = pagadores_mayor_importe_medio(gastos, 3)
    print("Los tres pagadores con un mayor importe medio en sus gastos son:")
    print(res)

def test_balance(gastos):
    print("El balance final de gastos es el siguiente (se muestran los usuarios
    por orden alfabético:)")
    res = balance(gastos)
    for usuario in sorted(res):
        print(usuario, "->", res[usuario])
```