

# Informes

## REPORTLAB

**Reportlab** es junto con **Pandas** los dos generadores de informes más utilizados en Python. Pandas se centra en elaborar informes en formato **HTML y Excel**, mientras que Reportlab se centra en la elaboración de informes en formato **PDF**.

Como práctica se elaborará un informe sencillo donde aparezca el listado de clientes. Empezaremos por como se crea y visualiza el pdf.

```
class Printer():  
  
    def reportCli():  
        try:  
            c = canvas.Canvas('informes/listadoclientes.pdf')  
            c.drawString(100,750,'Listado Clientes')  
            c.save()  
            rootPath = ".\\informes"  
            cont = 0  
            for file in os.listdir(rootPath):  
                if file.endswith('.pdf'):  
                    os.startfile("%s/%s" % (rootPath, file))  
                    cont = cont + 1  
  
        except Exception as error:  
            print('Error reporcli %s' % str(error))
```

- en primer lugar instalamos **reportlab**, librería que tiene todas las funciones necesarias.

- "c" es un nombre de variable tipo Canvas, que es el módulo que construye y soporta todo el entramado. Cuando llamamos a **c.Save()** finaliza el proceso de construcción y maquetación del pdf.

```
from reportlab.pdfgen import canvas  
import os
```

- tendremos una carpeta donde se guardarán todos los informes.

- **os.startfile** llamará al sistema para que busque el lector de pdf's por defecto, para lanzar el informe construido.

Con lo anterior se lanza un pdf. El resto de funciones a usar están explicadas en el manual de **ReportLab** que hay en el apartado *Recursos* del aula virtual.

## Partes de un informe

En la teoría, vimos que un informe podía tener hasta 5 partes, aunque no siempre tiene que ser así. En nuestro caso, plantearemos un mínimo de 3 partes para un informe sencillo y 4 en el caso de facturas (con el pie de informe).

### ○ Cabecera

Aquí suelen aparecer los datos generales de la empresa o institución, logo, fecha y otros posibles elementos que se repiten en los diferentes informes. Un ejemplo de código podría ser el siguiente:

```
def cabecera(self):
    try:
        logo = '..\\img\\logo.jpg'
        var.rep.setTitle('INFORMES')
        var.rep.setAuthor('Administración')
        var.rep.setFont('Helvetica', size=10)
        var.rep.line(45, 820, 525, 820)
        var.rep.line(45, 745, 525, 745)
        textcif = 'A0000000H'
        textnom = 'IMPORTACIÓN Y EXPORTACIÓN TEIS, S.L.'
        textdir = 'Avenida Galicia, 101 - Vigo'
        texttlfo = '886 12 04 64'
        var.rep.drawString(50, 805, textcif)
        var.rep.drawString(50, 790, textnom)
        var.rep.drawString(50, 775, textdir)
        var.rep.drawString(50, 760, texttlfo)
        var.rep.drawImage(logo, 450, 752)
```

Este módulo se llamará desde todos los informes ya que la cabecera de todos ellos será siempre la misma. En la llamada al logo, los valores de la posición hacen referencia a la esquina inferior izquierda de la imagen.

Desde el informe reportCli llamaremos a la cabecera de esta forma:

```
def reportCli():
    try:
        var.rep = canvas.Canvas('informes/listadoclientes.pdf', pagesize=A4)
        Printer.cabecera()
        textlistado = 'LISTADO DE CLIENTES'
        var.rep.drawString(255, 735, textlistado)
        var.rep.line(50, 728, 525, 728)
        var.rep.save()
        rootPath = ".\\informes"
```

**IMPORTANTE:** El diseño y maquetación de un informe, sobre todo, los valores numéricos que nos dan las posiciones consiste en la **simple técnica de ensayo prueba-error**.

### ○ Cuerpo

En el cuerpo se mostrará, en este caso, el listado de clientes de la empresa. No suelen incluirse todos los datos sino aquellos que solicite el cliente. En programas avanzados incluso, mediante un checkbox, puedes decidir cuales son los campos que se quieren incluir. En cuanto al código, se incluye la cabecera, una serie de líneas para mostrar los campos que queremos y luego una solicitud a la base de datos.

```
try:
    var.rep = canvas.Canvas('informes/listadoclientes.pdf', pagesize=A4)
    Printer.cabecera(self)
    var.rep.setFont('Helvetica-Bold', size=9)
    textlistado = 'LISTADO DE CLIENTES'
    var.rep.drawString(255, 735, textlistado)
    var.rep.line(45, 730, 525, 730)
    itemcli = ['Cod', 'DNI', 'APELLIDOS', 'NOMBRE', 'FECHA ALTA']
    var.rep.drawString(45, 710, itemcli[0])
    var.rep.drawString(90, 710, itemcli[1])
    var.rep.drawString(180, 710, itemcli[2])
    var.rep.drawString(325, 710, itemcli[3])
    var.rep.drawString(465, 710, itemcli[4])
    var.rep.line(45, 703, 525, 703)
    query = QSqlQuery()
    query.prepare('select codigo, dni, apellidos, nombre, fechalta from clientes order by apellidos, nombre')
    var.rep.setFont('Helvetica', size=10)
    if query.exec_():
        i = 50
        j = 690
        while query.next():
            var.rep.drawString(i, j, str(query.value(0)))
            var.rep.drawString(i+30, j, str(query.value(1)))
            var.rep.drawString(i+130, j, str(query.value(2)))
            var.rep.drawString(i+280, j, str(query.value(3)))
            var.rep.drawRightString(i+470, j, str(query.value(4)))
            j=j-30
        Printer.doc(textlistado)
    reportCli() → try → if query.exec_() → while query.next()
```

- **Pie de página o de informe**

En nuestro caso, haremos el pie de página. El pie de informe lo dejaremos para las *facturas* donde hay que realizar sumas de subtotales y totales.

En el pie de página suelen aparecer datos como: *número de página, nombre del informe o plantilla y fecha del informe*, pero también puede haber otros elementos dependiendo de la solicitud del cliente.

Un ejemplo podría ser el que se muestra a continuación:

```
def pie(textlistado):
    try:
        var.rep.line(50, 50, 525, 50)
        fecha = datetime.today()
        fecha = fecha.strftime('%d.%m.%Y %H.%M.%S')
        var.rep.setFont('Helvetica-Oblique', size=7)
        var.rep.drawString(460, 40, str(fecha))
        var.rep.drawString(275, 40, str('Página %s' % var.rep.getPageNumber()))
        var.rep.drawString(50, 40, str(textlistado))
    except Exception as error:
        print('Error en el pie de informe: %s' % str(error))
```

Y que tendremos que llamar desde el módulo principal del listado de clientes, es decir:

```
var.rep.drawString(i+280, j, str(quer
var.rep.drawString(i+420, j, str(quer
j=j-20
Printer.pie(textlistado)
var.rep.save()
rootPath = ".\\informes"
cont = 0
for file in os.listdir(rootPath):
```