**INFORME**

**PRÁCTICA 1**

**PTI**

Eduard González Moreno

Miguel Guerrero López

1/03/2018

# Entorno y configuración

La práctica se ha desarrollado en un entorno linux, para ello hemos tenido que utilizar apache2 como servidor web, y utilizado cgi-bin para correr scripts.

# Resolución

Para solucionar la práctica se ha utilizado dos archivos escritos en python. El primer archivos recoge los datos mediante get de un formulario que ya nos viene creado(carrental\_form\_new.html). Estos datos primeramente los tratamos para comprobar que nos vengan en el estado correcto.A continuación un fragmento de código en el cual recogemos los parámetro y comprobamos su validez, y en el caso de ser incorrecto devolvemos un error.

submodel = form.getvalue("sub\_model\_vehicle", "")  
 dieslloguer = form.getvalue("dies\_lloguer", "")  
 numvehi = form.getvalue("num\_vehicles", "")  
 des = form.getvalue("descompte", "")  
  
 dieslloguer = num(dieslloguer)  
 numvehi = num(numvehi)  
 des = flot(des)  
 if dieslloguer == -1:  
 print "Los dias de alquiler no contienen un numero"  
 elif dieslloguer <= 0:  
 print "Los dias de alquiler no pueden ser menor o igual a 0"  
 elif numvehi == -1:  
 print "El numero de vehiculos no contiene un numero"  
 elif numvehi <= 0:  
 print "El numero de vehiculos no puede ser menor o igual a 0"  
 elif des == -1:  
 print "El descuento no contiene un numero"  
 elif des < 0:  
 print "El descuento tiene que ser un numero igual o mayor que 0"

Las funciones que comprueban si los datos son números son las siguientes :

def num(s):  
 try:  
 return int(s)  
 except ValueError:  
 return -1

def flot(s):  
 try:  
 return float(s)  
 except ValueError:  
 return -1

Finalmente una vez todo es correcto realizamos el cálculo. Guardamos en un csv los datos para que en un futuro se pueda sacar una lista de todos los pedidos y por pantalla mostramos el desglose de su pedido y finalmente un total.

print "<h2>Su peticion de alquiler ha sido aceptada</h2><br><hr><br>"  
 print "<h3>Factura simplificada:</h3>"  
 print "Precio del modelo= "+model+"<br>"  
 print "Suplemento por submodelo= "+str(aumento)+"<br>"  
 print "Dias de alquiler= "+ str(dieslloguer) +"<br>"  
 print "Numero de vehiculos= "+ str(numvehi)+"<br>"  
 print "Descuento= "+str(des)+"<br>"  
 print "<hr>"  
 total = (num(model)\*flot(aumento)\*num(dieslloguer)\*num(numvehi))\*((100-des)/100)  
 print "<b>Total :</b>"+str(total)+"<br><br>"  
print "<a href='http://localhost/p1/carrental\_form\_new.html' > Volver </a>"

La segunda parte de la práctica, se ha realizado en otro fichero, como premisas nos pedía que se tenía poder identificar, por lo tanto nuestro script debía de ser capaz de recoger los parametros por post y validarlos.

Al no tener una BBDD donde comparar hemos la validación de contraseña en el mismo código.

if form['password'].value == "admin" and form['userid'].value == "admin":

…

else

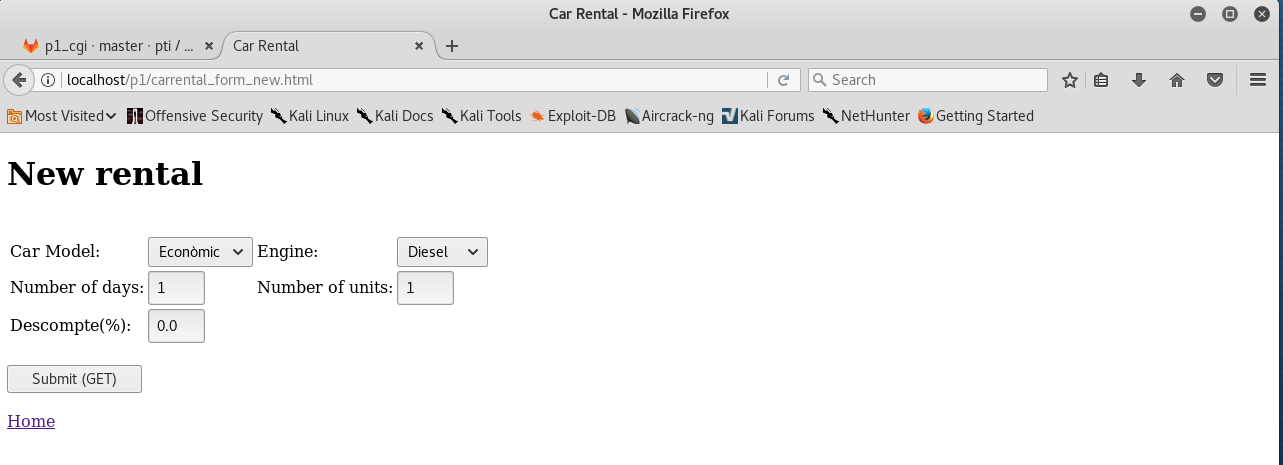
…

Una vez validado que los datos de autentificación son correctos mostramos por pantalla un listado de todas las solicitudes que se han realizado. A continuación el fragmento de codigo que se encarga de esta tarea

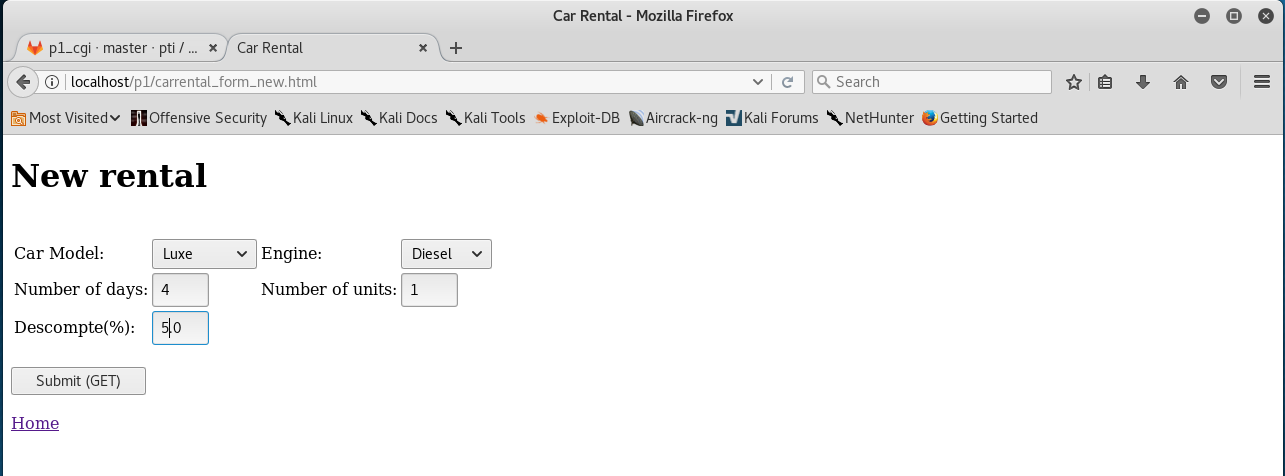
cr = csv.reader(open("BBDD.csv","rb"))  
 print "<table>"  
 print "<tr>"  
 print "<th> Modelo </th>"  
 print "<th> Submodelo </th>"  
 print "<th> Dias de alquiler </th>"  
 print "<th> Numero de vehiculos </th>"  
 print "<th> Descuento </th>"  
 print "</tr>"  
   
   
 for row in cr:   
 print "<tr>"  
 print "<td>" + row[0] + "</td>"  
 print "<td>" + row[1] + "</td>"  
 print "<td>" + row[2] + "</td>"  
 print "<td>" + row[3] + "</td>"  
 print "<td>" + row[4] + "</td>"  
 print "</tr>"  
 print "</table>"

# Evaluación de la solución

Para evaluar la solución, hemos utilizado nuestro navegador para comprobar que nuestro código funciona correctamente.



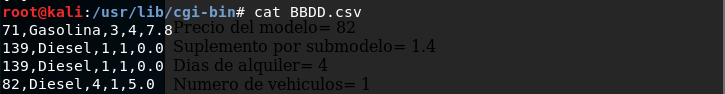
Este es el formulario que hay que rellenar para realizar un nuevo alquiler. A continuación vamos a realizar una prueba. Vamos a llenar el formulario con los siguientes datos:



Todo seguido hacemos el submit. A continuación se nos redirige a la página donde se nos muestra la petición de alquiler que hemos hecho anteriormente y comprobamos que coincide con los datos que hemos introducido en el formulario. También nos muestra el precio final de nuestro alquiler.

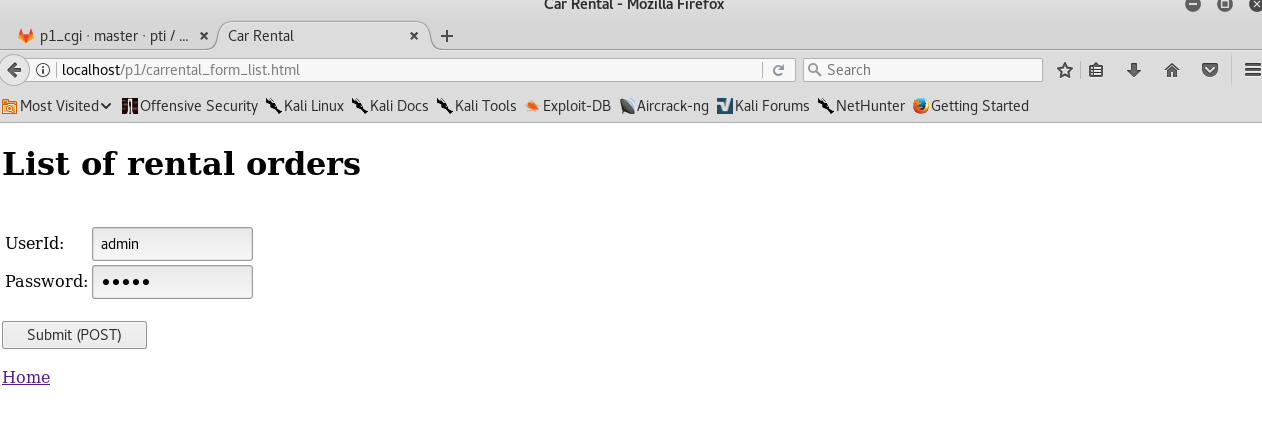


Como podemos apreciar el resultado es correcto. Como también tenemos que comprobar que el alquiler se guarde en el csv, para comprobarlo simplemente hacemos un cat sobre la BBDD y vemos que una nueva línea se ha añadido.



Como podemos ver la línea ha sido guardada correctamente.

Para comprobar el list.py hemos hecho lo siguiente:



Rellenamos el formulario, introducimos admin en UserId y admin en Password. A continuación pulsamos el botón de submit.



Y este es el resultado esperado. Nos aparece una tabla con los datos que tiene la BBDD. Como podemos ver en la imagen anterior, los datos coinciden con los que tiene el csv. Así pues queda demostrada la validez de nuestra solución.

# Ventajas de tecnologías

Como servidor web hemos utilizado un apache2, esta tecnología tiene la ventaja que es muy fácil de configurar y de utilizar, por otra parte es open source y es gratuito. La portabilidad de el servidor es muy fácil ya que es compatible con muchos más sistemas.

Otra de las tecnologías utilizadas ha sido python orientado a web. Al ser un lenguaje de scripts es muy fácil de realizar acciones de comprobaciones y montar pequeños procesos. Pero en parte es más complicado a la hora de agregar estilos al html con librerías externas estilo bootstrap.