

# TALLER:

## *“Proyectos Web en Python”*

django



Jorge Díaz M.  
Jorge.diazma@sansano.usm.cl  
Segundo Semestre 2019

# Sesión 5

## Sesiones y Formularios Parte 1

Formularios HTML, datos por POST y  
sesiones

# Sesión 5.1



**Formularios**

HTML



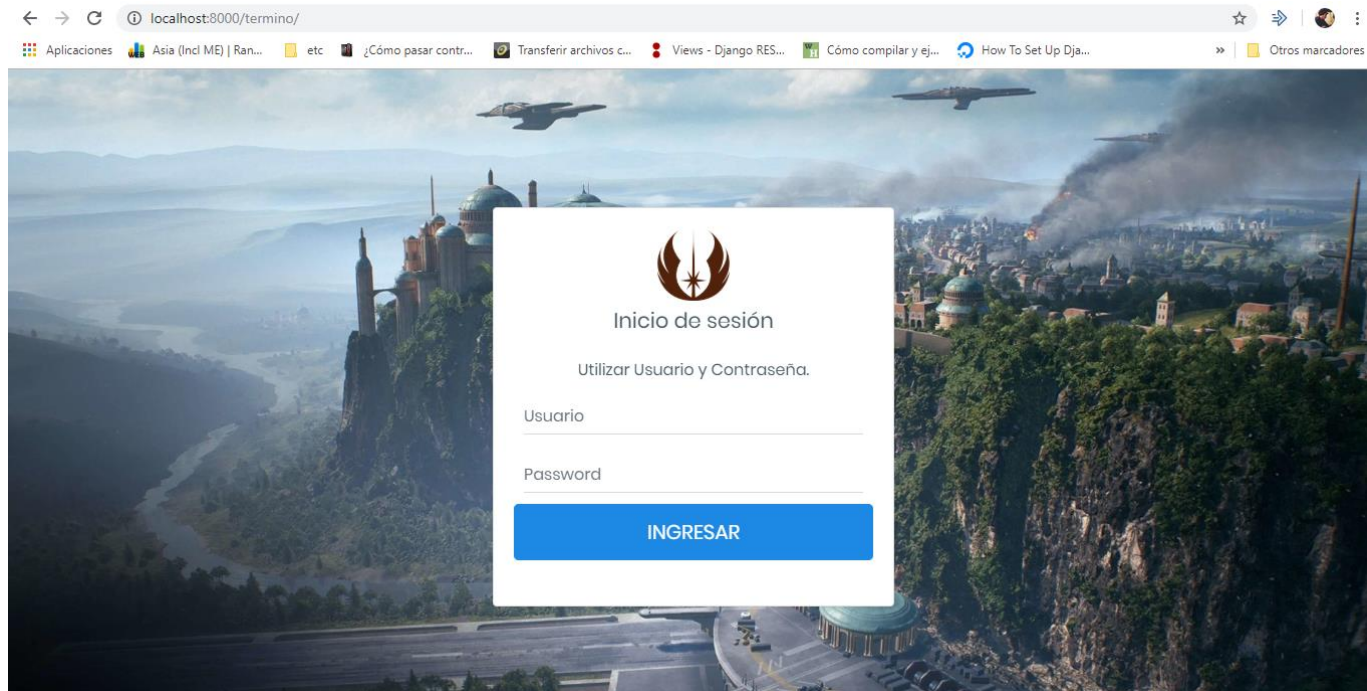
## Situación

### Situación de ejemplo

Imagine la siguiente situación: Queremos implementar el login en el sistema de talleres (de la semana pasada) al que le hicimos el modelo.



# HTML





# HTML

```
login.html
<!-- ===== -->
<section id="wrapper">
  <div class="login-register" style="background-image:url('/media/naboo.jpg')">
    <div class="login-box card">
      <div class="card-body">
        <form class="form-horizontal form-material" role="form" method="post">
          <center>
            <div style="padding:0px" class="upper">
              
            </div>
          </center>
          <center>
            <h3 style="padding:5px" class="box-title m-b-20">Inicio de sesión</h3>
            <h5>Utilizar Usuario y Contraseña.</h5>
          </center>
          <div id="panel-log" class="panel panel-default">
            <div class="panel-body">
              <div style="padding:10px" class="row-frn">
                <div class="input-group">
                  <span class="input-group-addon"><i class="glyphicon glyphicon-user"></i></span>
                  <input type="text" name="username" placeholder="Usuario" class="form-control" required="" id="id_username">
                </div>
              </div>
              <div style="padding:10px" class="row-frn">
                <div class="input-group">
                  <span class="input-group-addon"><i class="glyphicon glyphicon-lock"></i></span>
                  <input type="password" name="password" placeholder="Password" class="form-control" required="" id="id_password">
                </div>
              </div>
              <div class="form-group text-center m-t-2 0">
                <div class="col-xs-12">
                  <button class="btn btn-info btn-lg btn-block text-uppercase waves-effect waves-light" type="submit">Ingresar</button>
                </div>
              </div>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
</section>
```

Form

Botón de submit



# HTML

```
</center>
<center>
<h3 style="padding:5px" class="box-title m-b-20">Inicio de sesión</h3>
<h5>Utilizar Usuario y Contraseña.</h5>
</center>
<div id="panel-log" class="panel panel-default">
  <div class="panel-body">
    <div style="padding:10px" class="row-frn ">
      <div class="input-group">
        <span class="input-group-addon"><i class="glyphicon glyphicon-user"></i></span>
        <input type="text" name="username" placeholder="Usuario" class="form-control" required=""
          id="id_username">
      </div>
    </div>
    <div style="padding:10px" class="row-frn">
      <div class="input-group">
        <span class="input-group-addon"><i class="glyphicon glyphicon-lock"></i></span>
        <input type="password" name="password" placeholder="Password" class="form-control"
          required="" id="id_password">
      </div>
    </div>
    <div class="form-group text-center m-t-2 0">
      {% csrf_token %}
      <div class="col-xs-12">
        <button class="btn btn-info btn-lg btn-block text-uppercase waves-effect waves-light"
          type="submit">Ingresar</button>
      </div>
    </div>
  </div>
</div>
</form>
```

← Agregamos el  
csrf\_token

# Sesión 5.2

The Django logo, which consists of the word "django" in a white, lowercase, sans-serif font, centered within a dark green rounded rectangle.

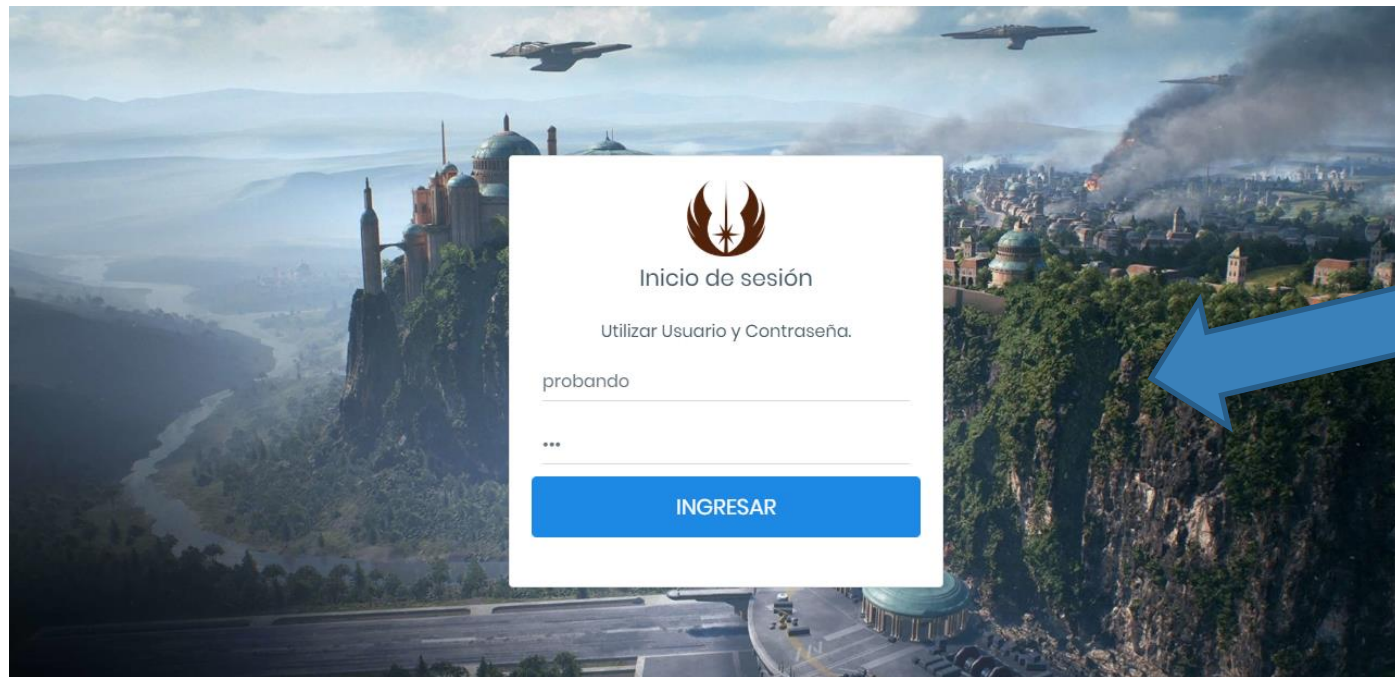
## Datos por POST

Recibiendo datos





## Pasando datos por post



**Escribimos  
valores y  
apretamos  
ingresar**



## Pasando datos por POST

```
System check identified no issues (0 silenced).  
September 26, 2019 - 12:13:19  
Django version 2.2.5, using settings 'taller.settings'  
starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.  
[26/Sep/2019 12:14:51] "POST /login HTTP/1.1" 200 4107
```



**Notamos que los datos  
si están llegando a la  
vista.**

**Ahora trabajaremos en  
al vista.**



## Pasando datos por POST

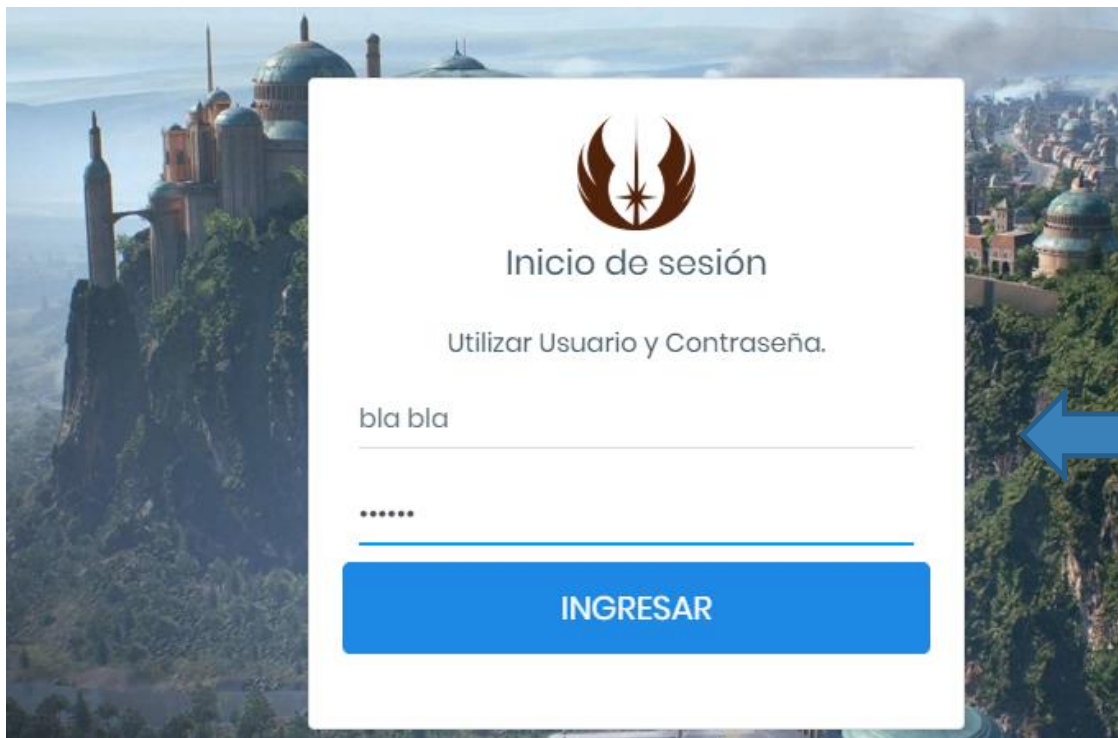
```
base.html x views.py x login.html x
1 from django.shortcuts import render, redirect
2 from django.contrib import auth
3 from django.contrib.auth.models import User
4 from django.contrib.auth.decorators import login_required
5
6
7 def login(request):
8     if request.method == 'POST':
9         if ("username" in request.POST.keys()) and ("password" in request.POST.keys()):
10             user = auth.authenticate(username=request.POST['username'], password=request.POST['password'])
11             if user is not None and user.is_active:
12                 auth.login(request, user)
13                 return redirect("/login/logueado/")
14             else:
15                 contexto={"error":"error"}
16                 return render(request,"login.html",contexto)
17         else:
18             contexto={"error":"error"}
19             return render(request,"login.html",contexto)
20     return render(request,"login.html")
21
22 @login_required(login_url='/login')
23 def vista_logueado(request):
24     return render(request,"logueado.html")
```

**Autenticamos**

**Si no, mandamos error.**



## Pasando datos por POST



A screenshot of the login screen from the game Jedi: Fallen Order. The background shows a floating island with a temple. In the center is a white login box with the Jedi Order symbol at the top. Below the symbol is the text 'Inicio de sesión' and 'Utilizar Usuario y Contraseña.'. There are two input fields: the first contains 'bla bla' and the second contains six dots. A blue 'INGRESAR' button is at the bottom.

Inicio de sesión

Utilizar Usuario y Contraseña.

bla bla

.....

INGRESAR



**No existe ese usuario**



## Pasando datos por POST

The image shows a login form for a Star Wars-themed application. The form is white and centered over a background image of a city with domes and mountains. The form contains the following elements:

- A Star Wars logo (a stylized bird-like head) at the top.
- The text "Inicio de sesión" (Login) below the logo.
- The text "Utilizar Usuario y Contraseña." (Use Username and Password.) below the login text.
- A text input field labeled "Usuario" (Username).
- A text input field labeled "Password".
- A red error message below the password field: "Usuario o clave incorrecta" (Username or password incorrect).
- A blue button labeled "INGRESAR" (Login) at the bottom.



No existe ese usuario



## Extra: Cerrar Sesión

```
{% extends 'base.html' %}
{% block muro %}
<center><h2>Usuario Logueado!</h2></center>
<form method="POST" role="form" action=".">
  {% csrf_token %}
  <input type="hidden" value="logout" name="cerrar">
  <center><button class="btn btn-info text-uppercase waves-effect waves-light" type="submit">Cerrar Sesión</button>
</form>
{% endblock %}
```

**Agregamos el botón para cerrar sesión y hacemos el formulario.**



## Extra: Cerrar Sesión

Usuario Logueado!

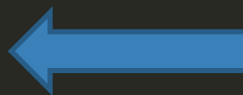
CERRAR SESIÓN

**Nos queda esto**



## Extra: Cerrar Sesión

```
22 @login_required(login_url='/login')
23 def vista_logueado(request):
24     if request.method == 'POST':
25         if ("cerrar" in request.POST.keys()):
26             auth.logout(request)
27             return redirect("/login/")
28     return render(request, "logueado.html")
29
```

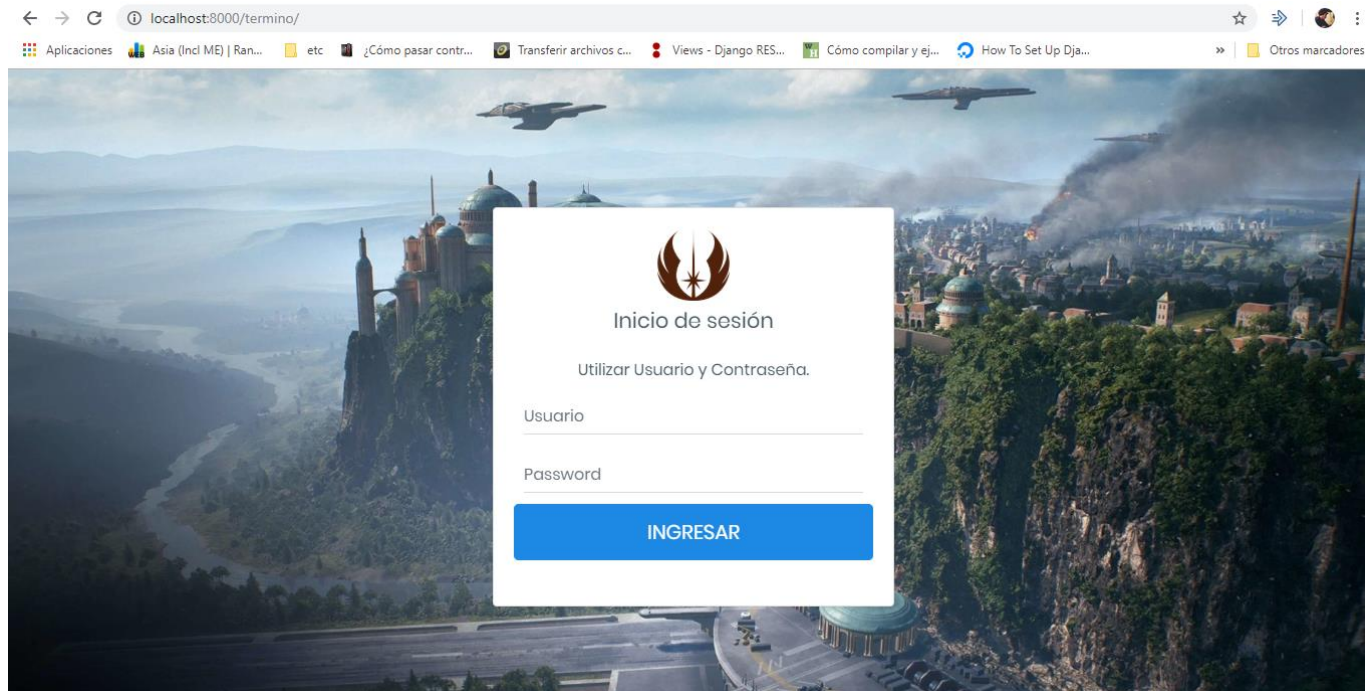


**Cerramos sesión en un comando.**





# Volvemos al inicio



# Sesión 5.3

**Buscar en la Base de datos**





## ¿Cuáles Field existen?

### Retornar una lista de resultados

<code>Pregunta.objects.all()</code>	Seleccionar todo
<code>Pregunta.objects.filter(enunciado='Hola')</code>	Seleccionar todas las preguntas cuyo enunciado sea "Hola"
<code>Pregunta.objects.exclude(enunciado='Hola')</code>	Seleccionar todas las preguntas cuyo enunciado sea distinto de "Hola"
<code>Pregunta.objects.all().order_by('enunciado')</code>	Ordenar por enunciado ascendentemente
<code>Pregunta.objects.all().order_by('-enunciado')</code>	Ordenar por enunciado descendientemente

### Retornar un único resultado

<code>Pregunta.objects.get(pk=1)</code>	Seleccionar pregunta cuya <i>Primary Key</i> (pk) sea 1
---	---

Todos los Tipos de modelos <https://docs.djangoproject.com/en/2.2/topics/db/queries/>



# Modelos de Datos

```
models.py
1 from django.db import models
2
3 class Relator(models.Model):
4     nombre=models.CharField(max_length=100, verbose_name="Nombre",blank=False,null=True)
5     rut=models.CharField(max_length=15, verbose_name="RUT",blank=True,null=True)
6     correo=models.EmailField(max_length=60, verbose_name="Correo electronico",blank=True,null=True)
7
8 class Estudiante(models.Model):
9     nombre=models.CharField(max_length=100, verbose_name="Nombre",blank=False,null=True)
10    rol=models.CharField(max_length=15, verbose_name="Rol USM",blank=True,null=True)
11    correo=models.EmailField(max_length=60, verbose_name="Correo electronico",blank=True,null=True)
12
13 class Taller(models.Model):
14     nombre=models.CharField(max_length=100, verbose_name="Nombre",blank=False,null=True)
15     cupos=models.IntegerField(verbose_name="Cupos",blank=True,null=True)
16     relator=models.ForeignKey(Relator, on_delete=models.CASCADE,blank=True,null=True)
17
18 class Inscripcion(models.Model):
19     taller=models.ForeignKey(Taller, on_delete=models.CASCADE,blank=True,null=True)
20     estudiante=models.ForeignKey(Estudiante, on_delete=models.CASCADE,blank=True,null=True)
21
```



## Actividad Evaluada 4

1. Elabore una página de login.
2. Haga el formulario en HTML de login.
3. Implemente el inicio de sesión.
4. Implemente un buscador de algún modelo pasando valores por POST.
5. Devuelva el resultado de la búsqueda en algún HTML.
6. Suba su proyecto comprimido a Moodle.



## Extra

### ¿Cómo manejo datos en una vista?

Les mostraré un proyecto real proyectado.





# Gracias por venir!

**¿Preguntas?**

La próxima clase aprenderemos a hacer formularios en django 😊

**jorge.diazma@sansano.usm.cl**