



fcfm

Ingeniería Industrial
FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE



Auxiliar 4

IN3501 - Tecnologías de Información y Comunicaciones para la Gestión

Auxiliares

José Canto - jose.canto@ug.uchile.cl

Javiera Ovalle - javiera.ovallet@gmail.com

Jorge Pinto - jorgepintoriveros@gmail.com

Agenda

- Resumen formularios
- Tipos de formularios
- Recuperando los datos
- Usando los datos
- Desafío 3



Resumen: Entorno Virtual



Para **crear** su entorno virtual:

```
virtualenv nombre_entorno_virtual
```

```
virtualenv nombre_entorno_virtual -p python3
```

Para **activar** su entorno virtual:

```
nombre_entorno_virtual\Scripts\activate
```

```
source nombre_entorno_virtual/bin/activate
```

Requirements.txt:

```
pip freeze > requirements.txt
```

```
pip install -r requirements.txt
```

Resumen: Django



Para **crear un proyecto** de Django e **ingresar**:

```
django-admin startproject  
nombre_proyecto
```

```
cd nombre_proyecto
```

Crear un **proyecto** y una **aplicación**:

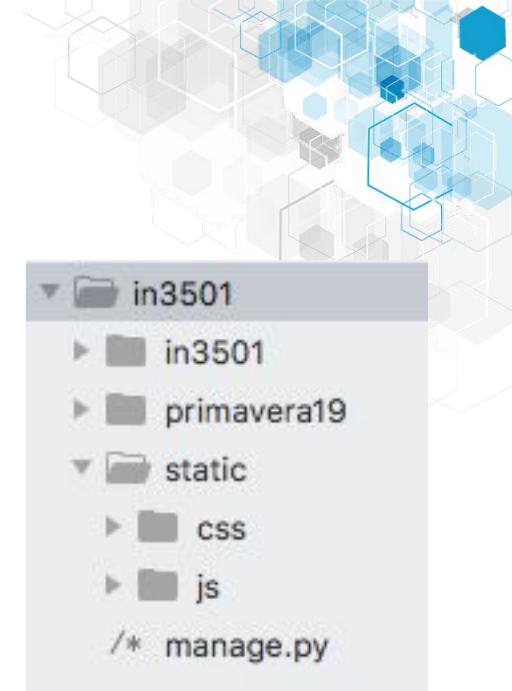
```
python manage.py startapp nombre_aplicacion
```

Iniciar el servidor:

```
$ python manage.py runserver
```

<http://localhost:8000> o <http://127.0.0.1:8000>

Resumen: Bootstrap



Debemos descargarlo desde [aquí](#)

Agregaremos las 2 carpetas al directorio /nombre/static/

En el archivo /nombre/nombre/settings.py debemos agregar

```
121     STATIC_URL = '/static/'  
122     STATICFILES_DIRS = [  
123         os.path.join(BASE_DIR, 'static'),  
124     ]
```

Importar CSS y JS en templates

```
<script src="{% static 'js/bootstrap.min.js' %}"></script>
```

```
<link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">
```

Resumen Formularios



Son la forma en que los usuarios interactúan con el sistema, se envía información

```
<form action="" method="POST">  
    ... inputs...  
</form>
```

“action”: Es a la url donde se envía la información

“method”:

POST: se envían los datos de manera invisible al usuario

GET: se envia a traves de la URL

Resumen Formularios



Se escriben dentro del cuerpo del HTML, y cuentan con:

- División de clase: Formulario
- Label: Etiqueta del campo que ve el usuario
- Definición de datos: Tipo - Clase - Id - Placeholder - name

```
<form method="POST" action="/add_proyecto"> {% csrf_token %}

    <div class="form-group">
        <label for="exampleInputTexto">Descripción del proyecto</label>
        <input type="text" class="form-control" id="exampleText"
            placeholder="Ingrese texto" name="descripcion">
    </div>

</form>
```

Tipos de formularios: texto y text área

```
<form action="#" autocomplete="off">

<div class="form-group">
  <label for="exampleInputEmail1">Type: text</label>
  <input type="text" class="form-control" id="exampleText" placeholder="Ingrese texto" name="inputText">
</div>

<div class="form-group">
  <label for="exampleFormControlTextarea">Type: textarea</label>
  <textarea class="form-control" id="exampleFormControlTextarea" rows="3" name="inputTextArea"></textarea>
</div>
```

Tipos de formularios: número y fecha

```
<div class="form-group">
  <label for="exampleNumber">Type: number</label>
  <input type="number" class="form-control" id="exampleNumber" placeholder="Ingrese numero"
  name="inputNumber">
</div>

<div class="form-group">
  <label for="exampleNumber">Type: date</label>
  <input type="date" class="form-control" id="exampleDate" placeholder="Ingrese Fecha" name="inputDate">
</div>
```

Tipos de formularios: Correo y password

```
<div class="form-group">  
  <label for="exampleInputEmail1">Type: email</label>  
  <input type="email" class="form-control" id="exampleEmail" placeholder="Ingrrese email" name="inputEmail">  
</div>  
  
<div class="form-group">  
  <label for="exampleInputPassword1">Type: password</label>  
  <input type="password" class="form-control" id="examplePassword" placeholder="ingrese contraseña" name="inputPassword">  
</div>
```

Tipos de formularios: Check box

```
<div class="form-group">

    <div class="form-check form-check-inline" >
        <input type="checkbox" class="form-check-input" id="exampleCheck1" name="inputCheck1">
        <label class="form-check-label" for="exampleCheck1">Check1</label>
    </div>

    <div class="form-check form-check-inline" >
        <input type="checkbox" class="form-check-input" id="exampleCheck2" name="inputCheck2">
        <label class="form-check-label" for="exampleCheck2">Check2</label>
    </div>
</div>
```

Tipos de formularios: Select lista

```
<div class="form-group">  
  
    <label for="exampleFormControlSelect1">Type: select</label>  
    <select class="form-control" id="exampleSelect" name="input Select">  
        <option value="1">Opcion 1</option>  
        <option value="2">Opcion 2</option>  
        <option value="3">Opcion 3</option>  
        <option value="4">Opcion 4</option>  
        <option value="5">Opcion 5</option>  
    </select>  
  
</div>
```

Tipos de formularios: Select radio

```
<div class="form-check">  
  <input class="form-check-input" type="radio" name="exampleRadios" id="exampleRadios1" value="option1" checked>  
  <label class="form-check-label" for="exampleRadios1"> Radio 1 </label>  
</div>  
<div class="form-check">  
  <input class="form-check-input" type="radio" name="exampleRadios" id="exampleRadios2" value="option2">  
  <label class="form-check-label" for="exampleRadios2"> Radio 2 </label>  
</div>  
</div>
```

Tipos de formularios: enviar

```
<div class="form-group">  
  <button type="submit" class="btn btn-primary">Enviar</button>  
</div>  
  
</form>
```

Recuperando los datos

Definimos una función en VIEWS.PY y utilizamos el NAME de los formularios

Usamos **request.POST["name"]** y lo asignamos a una variable

```
def add_proyecto(request):
    #recuperar la información que ingresó el cliente
    descripcion = request.POST['descripcion']
    cliente = request.POST['cliente']
    fecha = request.POST['fecha_límite']
    valores=request.POST['inputCheck']
    arch=request.POST['archivo']
    contraseña=request.POST['inputPassword']
```

Para guardar los datos en la BD (próximas auxiliares) se usa:

```
#agregar este registro a la base de datos
proyecto = Proyecto(descripcion=descripcion, cliente=cliente, fecha_límite=fecha)
proyecto.save()
```

Usando los datos

Podemos definir un diccionario en nuestra view, donde almacenaremos los datos y los podremos llevar a cualquier template

- Creamos un diccionario vacío: `contexto={}`
- Agregamos datos al diccionario: `contexto[llave]=informacion`

Ejemplo:

`contexto={}`

`contexto["descrip"]=descripcion`

`contexto["cliente"]=cliente`

`contexto["date"]=fecha`

```
def add_proyecto(request):
    #recuperar la información que ingresó el cliente
    descripcion = request.POST['descripcion']
    cliente = request.POST['cliente']
    fecha = request.POST['fecha_limite']
    valores=request.POST['inputCheck']
    arch=request.POST['archivo']
    contraseña=request.POST['inputPassword']
```

Usando los datos

Ejemplo:

diccionario={}

diccionario[“descrip”]=descripcion

diccionario[“cliente”]=cliente

diccionario[“date”]=fecha

```
def add_proyecto(request):
    #recuperar la información que ingresó el cliente
    descripcion = request.POST['descripcion']
    cliente = request.POST['cliente']
    fecha = request.POST['fecha_limite']
    valores=request.POST['inputCheck']
    arch=request.POST['archivo']
    contraseña=request.POST['inputPassword']
```

Y luego finalizamos la función **add_proyecto** con:

return render(request, “app/mostrar_datos.html”, diccionario)

Usando los datos



Ahora podemos usar estos datos en un template

Desde views.py tenemos:

```
def add_proyecto(request):
    ... recuperado la información...
    ... creación del diccionario...
    return render(request, "app/mostrar_datos.html", diccionario)
```

En el template llamamos al diccionario, y accedemos a los datos según su llave:

Usando los datos



```
def recuperar(request):
    texto=request.POST["inputText"]
    texto_largo=request.POST["textA"]
    diccionario={}
    diccionario["comentario"]=texto
    diccionario["comentario2"]=texto_largo
    return render(request, "mostrar_resultado.html", diccionario)
```

Para acceder al dato recuperado, se hace referencia al contenido del mismo, a través de su llave:

{nombre_llave}

```
{% extends 'navbar.html' %}

{% block title %}
Resultados
{% endblock title %}

{% block content %}

<h1> El texto ingresado por el usuario en el formulario fue: <h1>
{{comentario}}
<h1> El texto largo ingresado por el usuario en el formulario fue: <h1>
{{comentario2}}


{% endblock content %}
```

Desafío 3

A partir de lo aprendido, cree un formulario de inscripción a una actividad conjunta entre el DII, DCC e IQBT. Para esto deberá solicitar:

- Nombre del estudiante
- Edad
- Correo electrónico
- Seleccionar carrera (DII, DCC o IQBT)
- Tema preferido para la actividad (Animales, Deportes o Escritores)
- Opciones de comida (Bebestible, snacks y/o cena)

Recuerde que debe utilizar un tipo de formulario distinto para cada punto

Además, una vez enviada la información, deberá mostrar una página que **resuma todo lo ingresado por el estudiante.**

ENTREGAR VÍA U-CURSOS A MÁS TARDAR EL MIÉRCOLES 11/9 A LAS 23:59



fcfm

Ingeniería Industrial
FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE



Auxiliar 4

IN3501 - Tecnologías de Información y Comunicaciones para la Gestión

Auxiliares

José Canto - jose.canto@ug.uchile.cl

Javiera Ovalle - javiera.ovallet@gmail.com

Jorge Pinto - jorgepintoriveros@gmail.com