



INGENIERIA INDUSTRIAL
UNIVERSIDAD DE CHILE

Auxiliar 8

IN3501 Tecnologías de Información para la Comunicación y Gestión

Auxiliares:

Pablo Hernández – p.hernandez.m@ug.uchile.cl

Javiera Ovalle – javiera.ovallet@gmail.com

Jorge Pinto – jorgepintoriveros@gmail.com

Agenda

- Repaso consultas
- Consultas en Django
- Formularios en forms.py
- Formularios y bases de datos



Repositorio del ejemplo

- Comando:

```
Git clone https://github.com/IN3501/ejemplo\_base.git
```

- Trabajaremos con el proyecto llamado Aux_8

Repaso consultas

- En CMD: Acceder a la carpeta donde está manage.py y correr el comando

```
Python manage.py Shell
```

- En Command Line Client: Acceder a la terminal con el mismo nombre, y correr el comando

```
Use nombre_base;
```

Repaso consultas

INSERT INTO nombre_tabla **VALUES** (atributo1= valor 1, etc...)

Create

SELECT atr1, atr2,..., atrN **FROM** nombre_tabla **WHERE** condicion

Read

UPDATE nombre_tabla **SET** atr = nuevo_valor **WHERE** condicion

Update

DELETE FROM nombre_tabla **WHERE** condicion

Delete

Repaso consultas

- Consultas del tipo READ o SELECT sobre tabla Usuarios

Objetivo de la consulta	Sentencia en Django
Ver todos los datos	<code>Usuarios.objects.all()</code>
Filtrar un resultado por uno o más atributos	<code>Usuarios.objects.filter(atr1=valor, atr2=valor2)</code>
Ordenar un resultado filtrado	<code>Usuarios.objects.filter(atr=valor).order_by(atributo)</code>
Ordenar objetos	<code>Usuarios.objects.order_by("atributo")</code>
Ordenar objetos según más de un atributo	<code>Usuarios.objects.order_by("atr1", "atr2")</code>
Obtener un elemento específico para trabajar sobre él	<code>Usuarios.objects.get(atr=valor)</code>

Consultas en Django

- Ya vimos consultas del tipo READ
- También podemos hacer consultas de tipo CREATE, UPDATE y DELETE



Consultas Django CREATE

- Definimos una función que agregue los datos a la BD y le muestre un resultado al usuario. Usaremos `.save()`

```
def agregado(request):
```

```
    nuevo_usuario=Usuarios(nombre='Pepe', apellido='Cabezas',  
    direccion='beauchef 851', rut='123-4',  
    puntaje_psu='123', username='pepito',  
    password='pepe123', email='pepe@pepe.cl');
```

```
    nuevo_usuario.save()
```

```
    context={'Titulo':'Formulario Correcto', 'comentario':'Gracias'}  
    return render(request, 'ejemplo7/template.html', context)
```

Entidad
USUARIOS y sus
atributos

Guardar la
información en
la BD

Consultas Django UPDATE

- Para actualizar, podemos hacer una consulta para obtener objeto o trabajar sobre el nuevo usuario ya creado. Usaremos **.save()**

```
def modificar(request):
```

```
    persona=Usuarios.objects.get(id=1)  
    persona.direccion='Blanco Encalada 123'
```

```
    persona.save()
```

```
    context={'Titulo':'Información actualizada', 'comentario':'Gracias'}  
    return render(request, 'ejemplo7/template.html', context)
```

Obtener usuario
y darle nuevo
valor a su
dirección

Actualizar la
base de datos

Consultas Django DELETE

- También podemos eliminar datos, utilizando la función `.delete()`

```
def eliminar(request):  
  
    persona=Usuarios.objects.get(id=1)  
  
    persona.delete()  
  
    context={'Titulo':'El usuario ha sido eliminado', 'comentario':'Gracias'}  
    return render(request, 'ejemplo7/template.html', context)
```

Obtener el
usuario

Borrarlo de la BD

Formularios en forms.py

- En la misma carpeta donde se encuentra models.py, creamos el archivo llamado **forms.py**

```
from django import forms
```

```
class Formulario_NuevoUsuario(forms.Form):
```

```
    nombre = forms.CharField(label='Ingrese Nombre', widget=forms.TextInput(  
        attrs={'class': 'form-control',}))
```

```
    apellido = forms.CharField(label='Ingrese Apellido', widget=forms.TextInput(  
        attrs={'class': 'form-control',}))
```

```
    puntaje_psu = forms.IntegerField(label='Ingrese Puntaje PSU', widget=forms.TextInput(  
        attrs={'class': 'form-control', 'type': 'number'}))
```

```
    username = forms.CharField(label='Ingrese username', widget=forms.TextInput(  
        attrs={  
            'class': 'form-control',
```

Importamos
forms

Creamos los
campos para
completar

Creamos los
campos y
controlamos

Formularios en views.py

```
from .forms import Formulario_NuevoUsuario
from django.http import HttpResponseRedirect
from django.core.files.storage import FileSystemStorage

def Agregar(request):
    form=Formulario_NuevoUsuario()
    info={'titulo':'Agregar Usuarios','form':form}
    return render(request,'app_de_in3501/adduser.html', info)

def Agregado(request):
    new_user=Usuarios(nombre=request.POST['nombre'],apellido=request.POST['apellido'])
    usuario=Usuarios.objects.get(rut=request.POST['rut'])
    usuario.nombre="valentina"
    new_user.save()
    context={'Titulo':'Formulario Correcto', 'comentario':'Gracias'}
    return render(request, 'app_de_in3501/added.html', context)
```

Importamos el formulario y las clases de models.py

Llamamos al formulario para el template

Asignamos las respuestas del usuario a nuestro nuevo usuario y guardamos

Formularios en templates

- En el template **agregar.html** llamamos al formulario y lo enviamos a la view asociada a agregado

```
<div class="container text-center">
  <h3>{{titulo}}</h3>
  <p><em>{{intro}}</em></p>

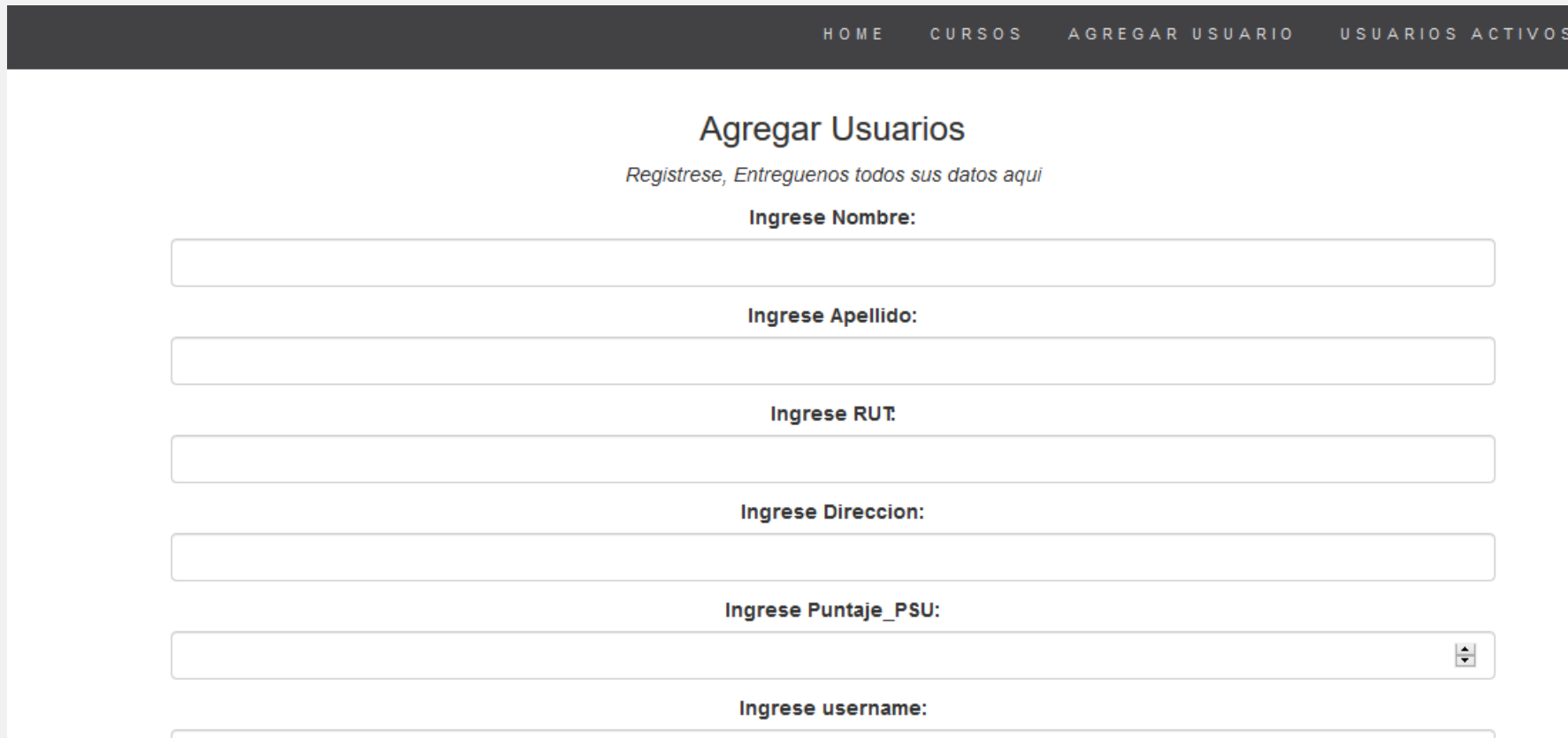
  <form action="/ejemplo7/agregado/" method="post" enctype="multipart/form-data">
    {% csrf_token %}
    {{form.as_p}}
    <button type="submit" class="btn btn-default">Registrar!</button>
  </form>

</div>
```

Llamamos al
formulario
para el
template

Formularios en templates

- Template agregar.html

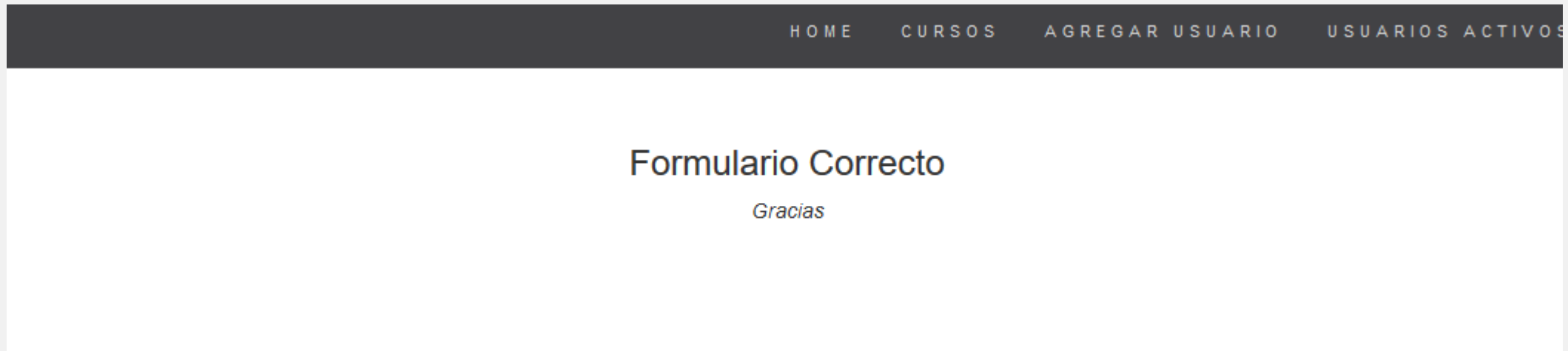


The screenshot displays a web form titled 'Agregar Usuarios' (Add Users) within a dark navigation bar. The navigation bar contains links: HOME, CURSOS, AGREGAR USUARIO, and USUARIOS ACTIVOS. The form itself is centered and contains the following elements:

- Header:** 'Agregar Usuarios' in bold, followed by the instruction 'Registrese, Entreguenos todos sus datos aqui' in italics.
- Fields:** A series of input fields, each preceded by a label:
 - Ingrese Nombre:** A single-line text input field.
 - Ingrese Apellido:** A single-line text input field.
 - Ingrese RUT:** A single-line text input field.
 - Ingrese Direccion:** A single-line text input field.
 - Ingrese Puntaje_PSU:** A single-line text input field with a small dropdown arrow icon on the right.
 - Ingrese username:** A single-line text input field (partially visible at the bottom).

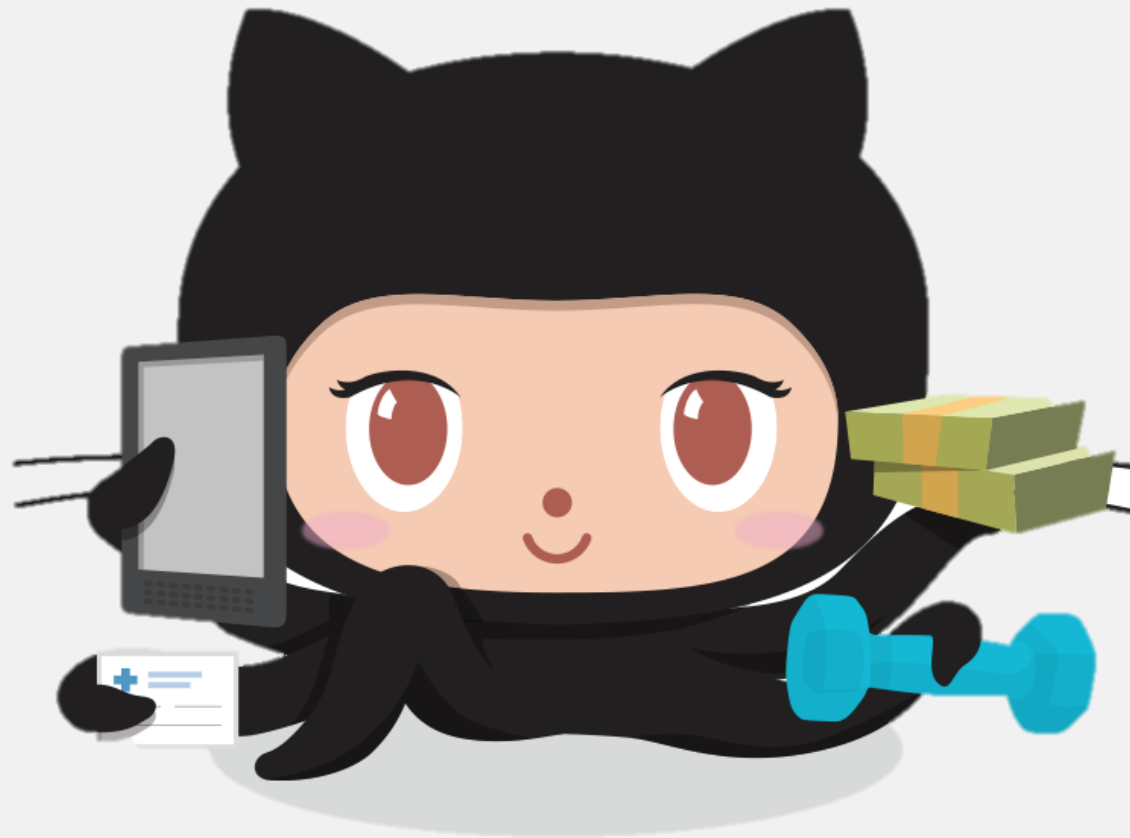
Formularios en templates

- Template agregado.html



Ejercicio en clase

- Cada respuesta puede llevarse o no una décima para el C3



Ejercicio en clase - Responda

- ¿Qué debería hacer en `views.py` si tengo más de un formulario definido en `forms.py`?

Ejercicio en clase - Responda

- ¿Qué debería hacer en **views.py** si tengo más de un formulario definido en forms.py?
- Si tomo la entidad **Usuarios**, ¿Cómo puedo actualizar el correo del usuario?

Ejercicio en clase - Responda

- ¿Qué debería hacer en **views.py** si tengo más de un formulario definido en forms.py?
- Si tomo la entidad **Usuarios**, ¿Cómo puedo actualizar el correo del usuario?
- Si tomo la entidad **Direccion**, ¿cómo puedo borrar la dirección 'Beauchef 850' de mi base de datos?

Ejercicio en clase - Responda

- ¿Qué debería hacer en `views.py` si tengo más de un formulario definido en `forms.py`?
- Si tomo la entidad `Usuarios`, ¿Cómo puedo actualizar el correo del usuario?
- Si tomo la entidad `Direccion`, ¿cómo puedo borrar la dirección 'Beauchef 850' de mi base de datos?
- ¿Cuáles son los `cuatro archivos` que debo modificar para que funcione mi formulario?

Ejercicio en clase - Responda

- ¿Qué debería hacer en **views.py** si tengo más de un formulario definido en forms.py?
- Si tomo la entidad **Usuarios**, ¿Cómo puedo actualizar el correo del usuario?
- Si tomo la entidad **Direccion**, ¿cómo puedo borrar la dirección 'Beauchef 850' de mi base de datos?
- ¿Cuáles son los **cuatro archivos** que debo modificar para que funcione mi formulario?
- ¿Qué **archivo debo modificar** y cómo para que una consulta me muestre el nombre de la persona y no objeto(1)?

Desafío

- Deberá crear un proyecto trabaje sobre la tabla CURSOS de la base IN3501 y que permita
 1. Agregar un nuevo curso
 2. Actualizar las UB de los cursos (revise diapositiva 6, al final)
 3. Borrar los curso que se impartan en primavera u otoño, según quiera el usuario
 4. Mostrar los cursos en una tabla separada según Otoño y Primavera

ENTREGA MIÉRCOLES 28 DE NOVIEMBRE A LAS 23:59

Debe tener en cuenta que los tutores probarán sus formularios y sus revisarán que las vistas cumplan con lo pedido. Si no compila de manera correcta, tendrá descuento de -3 puntos, y si no sube su carpeta completa, será evaluad@ con nota 1.0



INGENIERIA INDUSTRIAL
UNIVERSIDAD DE CHILE

Auxiliar 8

IN3501 Tecnologías de Información para la Comunicación y Gestión

Auxiliares:

Pablo Hernández – p.hernandez.m@ug.uchile.cl

Javiera Ovalle – javiera.ovallet@gmail.com

Jorge Pinto – jorgepintoriveros@gmail.com