

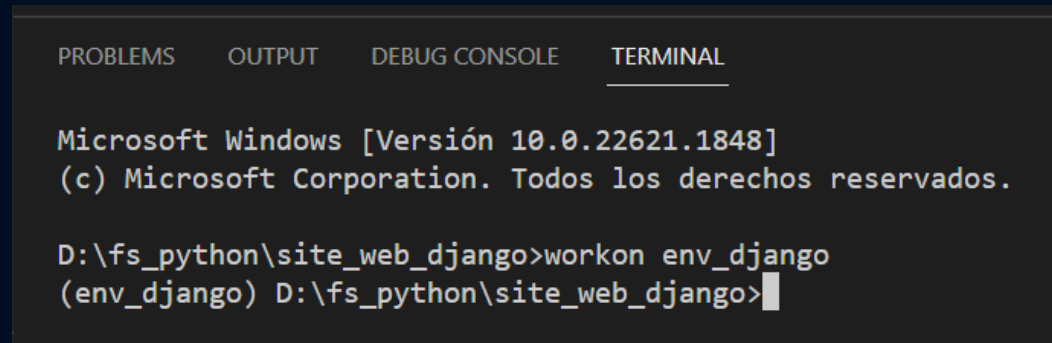
# DJANGO

AUTHENTICATION

Material de trabajo de Sesión 8 – Módulo 6

# EXERCISE 1: SISTEMA DE AUTENTICACIÓN DE DJANGO.

- 1) Abrir el proyecto en curso en su VSCode
- 2) Abrir el terminal y activar el ambiente correspondiente

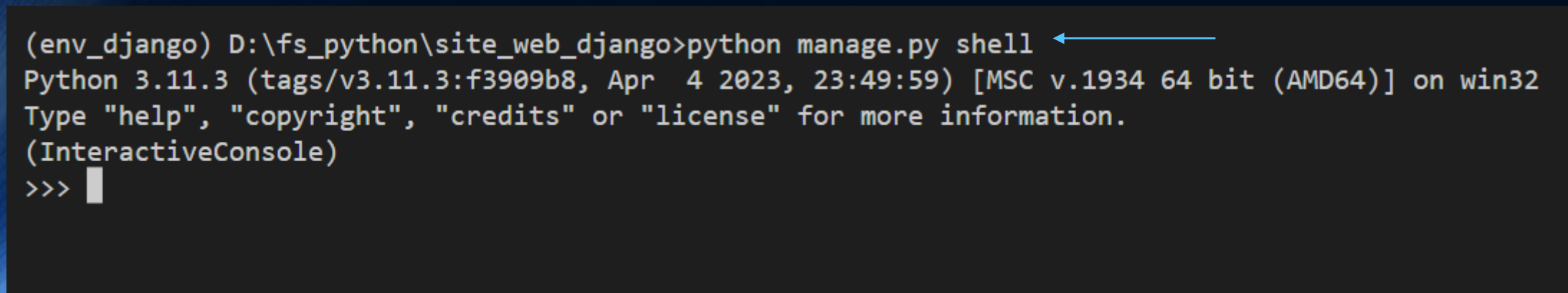


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Microsoft Windows [Versión 10.0.22621.1848]
(c) Microsoft Corporation. Todos los derechos reservados.

D:\fs_python\site_web_django>workon env_django
(env_django) D:\fs_python\site_web_django>
```

- 3) En el terminal, abrir la Shell de Django:



```
(env_django) D:\fs_python\site_web_django>python manage.py shell ←
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>>
```

# EXERCISE 1: SISTEMA DE AUTENTICACIÓN DE DJANGO.

4) Procedamos a crear un usuario. En la Shell escriba:

```
(env_django) D:\fs_python\site_web_django>python manage.py shell
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from django.contrib.auth.models import User
>>> user = User.objects.create_user('Carlos','carlosgarcia@correo.com','carlosgarcia')
>>> █
```

5) Cambiar contraseña al usuario Carlos:

```
>>>
>>> u = User.objects.get(username='Carlos')
>>> u.set_password('12345')
>>> u.save()
>>> █
```

Si alguno de  
estos pasos te  
falla, verifica la  
sintaxis

# EXERCISE 1: SISTEMA DE AUTENTICACIÓN DE DJANGO.

6) En esa misma sesión de la Shell, podemos observar los atributos de la instancia creada anteriormente:

```
>>> print(u.password)
pbkdf2_sha256$600000$E8HV53g4mmg9qLK7Yy4HCB$3eT3D9JTgGqIt1hwcf9opVkD0qlb1IOuwvhTa1o+HJM=
>>> print(u.email)
carlosgarcia@correo.com
>>> █
```

7) Procedamos a autenticar al usuario:

```
>>> from django.contrib.auth import authenticate
>>> user = authenticate(username='Carlos', password='12345')
>>> █
```

Si alguno de  
estos pasos te  
falla, verifica la  
sintaxis



# EXERCISE 1: SISTEMA DE AUTENTICACIÓN DE DJANGO.

8) Verifiquemos si el usuario se autenticó correctamente:

```
>>> from django.contrib.auth import authenticate
>>> user = authenticate(username='Carlos', password='12345')
>>> 
```

Si la autenticación es exitosa,

```
>>> print(user)
Carlos
>>> 
```

Veremos el username.

En caso contrario, Python devuelve None.

Para salir de la Shell, escriba: `exit()`

Si alguno de  
estos pasos te  
falla, verifica la  
sintaxis

# EXERCISE 2: CREACIÓN DE UNA PÁGINA DE REGISTRO DE USUARIOS.

1) Abrir el terminal e instalar el siguiente componente:

```
(env_django) D:\fs_python\site_web_django>pip install crispy-bootstrap5
```

Para verificar que se ha instalado correctamente:

```
(env_django) D:\fs_python\site_web_django>pip freeze
asgiref==3.7.2
beautifulsoup4==4.12.2
crispy-bootstrap5==0.7
Django==4.2.2
django-bootstrap-v5==1.0.11
django-crispy-forms==2.0
soupsieve==2.4.1
sqlparse==0.4.4
tzdata==2023.3

(env_django) D:\fs_python\site_web_django>
```

# EXERCISE 2: CREACIÓN DE UNA PÁGINA DE REGISTRO DE USUARIOS.

2) Registramos los nuevos componentes en el settings.py:

Así mismo, agregar los "templates packs": →

```
site_web_django > settings.py > ...  
33  INSTALLED_APPS = [  
34      'django.contrib.admin',  
35      'django.contrib.auth',  
36      'django.contrib.contenttypes',  
37      'django.contrib.sessions',  
38      'django.contrib.messages',  
39      'django.contrib.staticfiles',  
40      'boards',  
41      'bootstrap5',  
42      'crispy_forms',  
43      'crispy_bootstrap5',  
44  ]  
45  
46  CRISPY_ALLOWED_TEMPLATE_PACKS = "bootstrap5"  
47  CRISPY_TEMPLATE_PACK = "bootstrap5"  
48
```

# EXERCISE 2: CREACIÓN DE UNA PÁGINA DE REGISTRO DE USUARIOS.

3) Procedamos a crear la una nueva clase de formulario de registro, partiendo del modelo de usuario que contiene Django.

```
boards > forms.py > ...
1  from django import forms
2  from .models import BoardsModel
3  from django.contrib.auth.forms import UserCreationForm
4  from django.contrib.auth.models import User
5
6  > class InputForm(forms.Form): ...
11
12 > class BoardsForm(forms.ModelForm): ...
17
18 class RegistroUsuarioForm(UserCreationForm):
19     email = forms.EmailField(required=True)
20
21     class Meta:
22         model = User
23         fields = ("username", "email", "password1", "password2")
24
25     def save(self, commit=True):
26         user = super(RegistroUsuarioForm, self).save(commit=False)
27         user.email = self.cleaned_data['email']
28         if commit:
29             user.save()
30         return user
31
```



# EXERCISE 2: CREACIÓN DE UNA PÁGINA DE REGISTRO DE USUARIOS.

## 4) Ahora crearemos el nuevo template **registro.html**

```
templates > <> registro.html > ...
1  {% include "menu.html" %}
2  {% block content %}
3      {% load crispy_forms_tags %}
4      <!--Registro-->
5      <div class="container py-5">
6          <h1>Registro</h1>
7          <form method="POST">
8              {% csrf_token %}
9              {{ register_form|crispy }}
10             <button class="btn btn-primary" type="submit">Register</button>
11          </form>
12          <p class="text-center">
13              Si tienes actualmente una cuenta, inicie su sesión en
14              <a href="/login">login</a>.
15          </p>
16      </div>
17  {% endblock %}
18
```

# EXERCISE 2: CREACIÓN DE UNA PÁGINA DE REGISTRO DE USUARIOS.


## 5) Agregar ruta de registro al urls.py

Ese slash es para poder continuar en la línea siguiente

```
boards > 📁 urls.py > ...
1  from django.urls import path
2  from .views import IndexPageView, obtenerFecha, menuView, \
3     mostrar, datosform_view, formmanual_view, boardsform_view, registro_view
4
5  urlpatterns = [
6     path('', IndexPageView.as_view(), name = 'index'),
7     path('fecha/<name>', obtenerFecha, name='fecha'),
8     path('menu/', menuView, name='menu'),
9     path('mostrar/', mostrar, name = 'mostrar'),
10    path('datosform/', datosform_view, name = 'datosform'),
11    path('formmanual/', formmanual_view, name = 'formmanual'),
12    path('boardsform/', boardsform_view, name = 'boardsform'),
13    path('registro/', registro_view, name="registro"),
14 ]
15
```

# EXERCISE 2: CREACIÓN DE UNA PÁGINA DE REGISTRO DE USUARIOS.

## 6) Ahora, agregaremos la vista:

```
boards >  views.py > ...  
1  from django.shortcuts import render  
2  from django.http import HttpResponse , HttpResponseRedirect  
3  from django.views.generic import TemplateView  
4  import datetime  
5  from .forms import InputForm, BoardsForm, RegistroUsuarioForm  
6  from tokenize import PseudoExtras  
7  from django.contrib import messages  
8  from django.contrib.auth import login  
9
```

Nuevas importaciones

# EXERCISE 2: CREACIÓN DE UNA PÁGINA DE REGISTRO DE USUARIOS.

## 6) Ahora, agregaremos la vista:

boards > views.py > ...

```
70
71 def registro_view(request):
72     if request.method == "POST":
73         form = RegistroUsuarioForm(request.POST)
74         if form.is_valid():
75             user = form.save()
76             login(request, user)
77             messages.success(request, "Registrado Satisfactoriamente." )
78             return HttpResponseRedirect('/menu')
79
80     messages.error(request, "Registro invalido. Algunos datos ingresados no son correctos")
81     form = RegistroUsuarioForm()
82     context = { "register_form" : form }
83     return render(request, "registro.html", context)
84
```



# EXERCISE 2: CREACIÓN DE UNA PÁGINA DE REGISTRO DE USUARIOS.

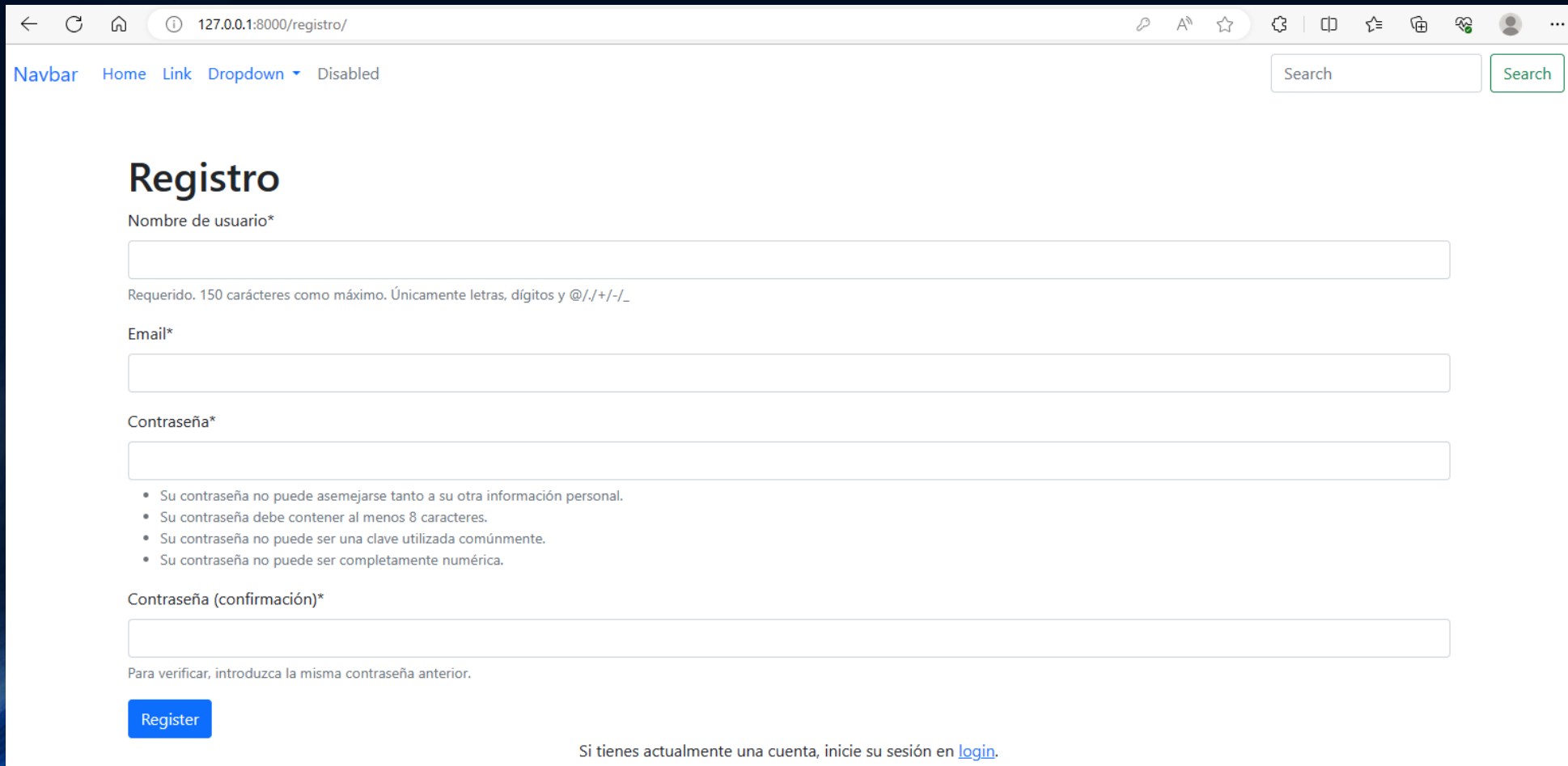
7) En el terminal, inicie el servidor:

```
(env_django) D:\fs_python\site_web_django>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 21, 2023 - 22:40:42
Django version 4.2.2, using settings 'site_web_django.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

# EXERCISE 2: CREACIÓN DE UNA PÁGINA DE REGISTRO DE USUARIOS.

8) En el navegador de su preferencia, ingrese a:



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/registro/`. The page features a navigation bar with links: [Navbar](#), [Home](#), [Link](#), [Dropdown](#) (with a dropdown arrow), and [Disabled](#). On the right side of the navbar is a search bar with a "Search" button. The main content area is titled "Registro" and contains the following form fields and instructions:

- Nombre de usuario\***: A text input field. Below it, the text reads: "Requerido. 150 caracteres como máximo. Únicamente letras, dígitos y @/./+/-/\_".
- Email\***: A text input field.
- Contraseña\***: A text input field. Below it, a list of password requirements is shown:
  - Su contraseña no puede asemejarse tanto a su otra información personal.
  - Su contraseña debe contener al menos 8 caracteres.
  - Su contraseña no puede ser una clave utilizada comúnmente.
  - Su contraseña no puede ser completamente numérica.
- Contraseña (confirmación)\***: A text input field. Below it, the text reads: "Para verificar, introduzca la misma contraseña anterior."

At the bottom left of the form is a blue "Register" button. At the bottom center, there is a link: "Si tienes actualmente una cuenta, inicie su sesión en [login](#)."

# EXERCISE 3: INICIO DE SESIÓN.

## 1) Creamos el template `login.html`

```
templates > <> login.html > ...
1  {% include "menu.html" %}
2  {% block content %}
3      {% load crispy_forms_tags %}
4      <!--Login-->
5      <div class="container py-5">
6          <h1>Login</h1>
7          <form method="POST">
8              {% csrf_token %}
9              {{ login_form|crispy }}
10             <button class="btn btn-primary" type="submit">Login</button>
11          </form>
12          <p class="text-center">
13              Si no tienes una cuenta, regístrate en <a href="/registro">, para crear una cuenta</a>.
14          </p>
15      </div>
16  {% endblock %}
17
```

# EXERCISE 3: INICIO DE SESIÓN.

## 2) Agregamos la URL

boards >  urls.py > ...

```
1  from django.urls import path
2  from .views import IndexPageView, obtenerFecha, menuView, \
3     mostrar, datosform_view, formmanual_view, boardsform_view, registro_view, login_view
4
5  urlpatterns = [
6     path('', IndexPageView.as_view(), name = 'index'),
7     path('fecha/<name>', obtenerFecha, name='fecha'),
8     path('menu/', menuView, name='menu'),
9     path('mostrar/', mostrar, name = 'mostrar'),
10    path('datosform/', datosform_view, name = 'datosform'),
11    path('formmanual/', formmanual_view, name = 'formmanual'),
12    path('boardsform/', boardsform_view, name = 'boardsform'),
13    path('registro/', registro_view, name="registro"),
14    path('login/', login_view, name="login"),
15 ]
16
```





# EXERCISE 3: INICIO DE SESIÓN.

## 3) Agregamos la vista

boards > views.py > ...

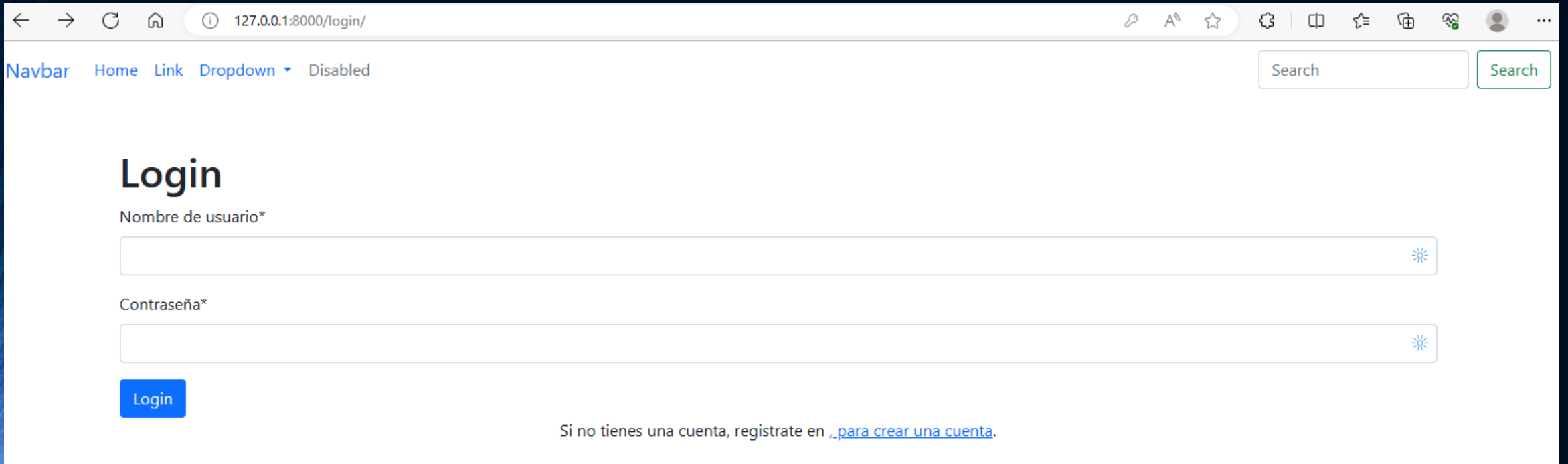
```
8 from django.contrib.auth import login, authenticate
9 from django.contrib.auth.forms import AuthenticationForm
10
```

boards > views.py > ...

```
86 def login_view(request):
87     if request.method == "POST":
88         form = AuthenticationForm(request, data=request.POST)
89         if form.is_valid():
90             username = form.cleaned_data.get('username')
91             password = form.cleaned_data.get('password')
92             user = authenticate(username=username, password=password)
93             if user is not None:
94                 login(request, user)
95                 messages.info(request, f"Iniciaste sesión como: {username}.")
96                 return HttpResponseRedirect('/menu')
97             else:
98                 messages.error(request, "Invalido username o password.")
99         else:
100             messages.error(request, "Invalido username o password.")
101
102     form = AuthenticationForm()
103     context = {"login_form": form}
104     return render(request, "login.html", context)
105
```

# EXERCISE 3: INICIO DE SESIÓN.

4) En el navegador de su preferencia, ingrese a:



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/login/`. The page has a navigation bar with links: [Navbar](#), [Home](#), [Link](#), [Dropdown](#) (with a dropdown arrow), and [Disabled](#). On the right side of the navbar, there is a search bar with the text "Search" and a green "Search" button. The main content area is titled "Login" in a large, bold font. Below the title, there are two input fields: "Nombre de usuario\*" and "Contraseña\*", both with a blue eye icon on the right side. Below the "Contraseña\*" field is a blue "Login" button. At the bottom of the page, there is a text link: "Si no tienes una cuenta, regístrate en [.para crear una cuenta.](#)".

¡Prueba con credenciales incorrectas y correctas (la creada en la Shell) para observar el comportamiento!

# EXERCISE 4: CIERRE DE SESIÓN DE UN USUARIO

1) Editamos el  
template **menu.html**

```
templates > <> menu.html > ...
35         <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
36         <button class="btn btn-outline-success" type="submit">Search</button>
37     </form>
38 </div>
39 </div>
40 <!-- Hasta la línea anterior el archivo no se modifica. Insertar: -->
41 <div class="collapse navbar-collapse" id="navbarText">
42     <ul class="navbar-nav mr-auto">
43         {% if user.is_authenticated %}
44         <li class="nav-item">
45             <a class="nav-link" href="/logout">Salir</a>
46         </li>
47         <li class="nav-item">
48             <a class="nav-link" href="#">Hola, {{user.username}}</a>
49         </li>
50         {% else %}
51         <li class="nav-item">
52             <a class="nav-link" href="/login">Login</a>
53         </li>
54         {% endif %}
55     </ul>
56 </div>
57 <!-- Hasta hasta acá la modificación -->
58 </nav>
59
60 {% endblock %}
```

# EXERCISE 4: CIERRE DE SESIÓN DE UN USUARIO

## 2) Agregamos la URL


```
boards >  urls.py > ...  
1  from django.urls import path  
2  from .views import IndexPageView, obtenerFecha, menuView, \  
3      mostrar, datosform_view, formmanual_view, boardsform_view, registro_view, login_view, logout_view  
4  
5  urlpatterns = [  
6      path('', IndexPageView.as_view(), name = 'index'),  
7      path('fecha/<name>', obtenerFecha, name='fecha'),  
8      path('menu/', menuView, name='menu'),  
9      path('mostrar/', mostrar, name = 'mostrar'),  
10     path('datosform/', datosform_view, name = 'datosform'),  
11     path('formmanual/', formmanual_view, name = 'formmanual'),  
12     path('boardsform/', boardsform_view, name = 'boardsform'),  
13     path('registro/', registro_view, name="registro"),  
14     path('login/', login_view, name="login"),  
15     path('logout/', logout_view, name="logout"),  
16  
17 ]
```







# EXERCISE 4: CIERRE DE SESIÓN DE UN USUARIO

## 3) Agregamos la vista

```
boards >  views.py > ...
```

```
7  from django.contrib import messages
8  from django.contrib.auth import login, authenticate, logout
9  from django.contrib.auth.forms import AuthenticationForm
10
```

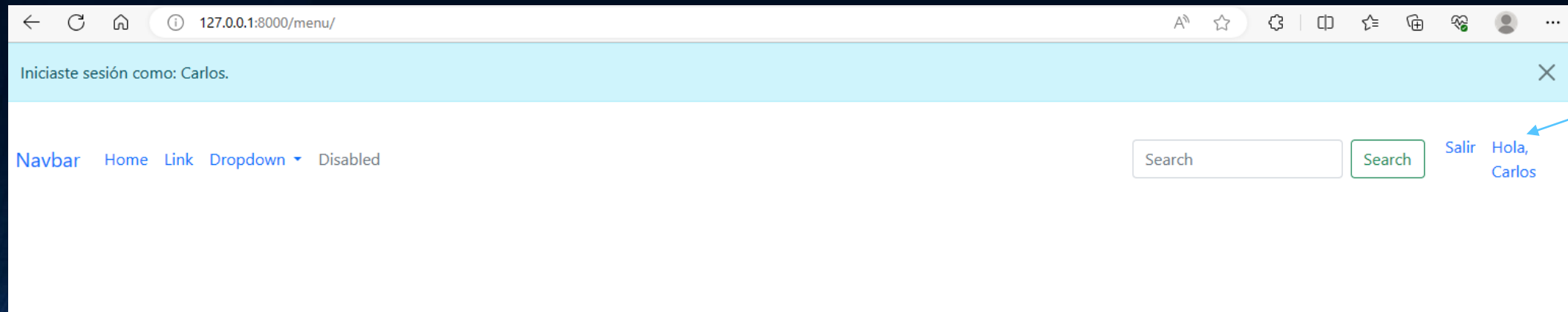


```
boards >  views.py > ...
```

```
105 def logout_view(request):
106     logout(request)
107     messages.info(request, "Se ha cerrado la sesión satisfactoriamente.")
108     return HttpResponseRedirect('/menu')
109
```

# EXERCISE 4: CIERRE DE SESIÓN DE UN USUARIO

4) En el navegador de su preferencia, inicia sesión con las credenciales adecuadas. Podrás observar lo siguiente:



Y al presionar "Salir":

