

EJERCICIO:

Para esta revisión práctica del sitio de administración de Django, se debe:

- Agregar en el menú la opción de Agregar Libro, la cual hace referencia a: <http://localhost:8000/inputbook/>

The screenshot shows a Django admin interface for a 'Books' model. The model has three fields: 'Title' (CharField), 'Author' (CharField), and 'Rating' (IntegerField). The 'Rating' field has a note indicating it must be between 0 and 10000. The 'Add' button is visible at the bottom right of the form.

Books

Title	Author	Rating
Placeholder	Placeholder	Placeholder

Add

Fields

Name	Type	Description
Title	CharField	Placeholder
Author	CharField	Placeholder
Rating	IntegerField	Placeholder Value must be between 0 and 10000

- Al seleccionar en el menú la opción Agregar Libros, ésta muestra:

Titulo:

Autor:

Valoracion: Valor entre 0 y 10000

Título:

Autor:

Valoración: Valor entre 0 y 10000

Enviar

```
book > 🐍 views.py > ⚒ inputbook
 1  from django.shortcuts import render, redirect    La importación "django" no se
 2  from .models import BookModel
 3  from .forms import BookModelForm
 4
 5  def index(request):
 6      context = {}
 7      return render(request, "index.html", context)
 8
 9  def listbook(request):
10      libros = BookModel.objects.all()
11      context = {"libros": libros}
12      return render(request, "listbook.html", context)
13
14  def inputbook(request):
15      if request.method == "POST":
16          form = BookModelForm(request.POST)
17          if form.is_valid():
18              form.save()
19              return redirect("index")
20      else:
21          form = BookModelForm()
22      context = {"form": form}
23      return render(request, "inputbook.html", context)
```

```
book > 🐍 forms.py > 📄 BookModelForm > 📄 Meta
 1  from django import forms    La importación "django" no se
 2  from .models import BookModel
 3
 4  class BookModelForm(forms.ModelForm):
 5      class Meta:
 6          model = BookModel
 7          fields = ['titulo', 'autor', 'valoracion']
```

- Agregar en la plantilla que renderiza el formulario de agregar libros las opciones del menú, de manera que se observe así:

The screenshot shows a Django application interface. At the top, there's a navigation bar with links: Site Web Django, Inicio, Agregar Libros, Libros, Acerca de, and Contacto. On the right side of the header is a "Login" button. Below the header, there's a form for adding a book with fields for Title, Author, and Rating (with a note saying it must be between 0 and 10000). A "Enviar" button is at the bottom of the form. Below the form is a horizontal line separating the content from the footer. The footer contains three sections: DESCRIPCIÓN (with the text "Desarrollo de Sitio Web de Django."), PRODUCTOS (with a link to "Libros"), and CONTACTO (with address, email, and phone number information). At the very bottom of the page is a copyright notice: "© 2022 Copyright: SiteWebDjango.com".

This screenshot shows the same Django application as the previous one, but with different styling. The "Enviar" button is now highlighted in blue. The overall layout is similar, with the navigation bar, form, and footer sections all present. The footer's CONTACTO section includes small icons next to the text.

- Restringir la aparición del menú, para que solo aparezca si el usuario ha iniciado sesión.

```
book > templates > registration > login.html
1  <h3 class="text-center my-4">Ingrese a su cuenta:</h3>
2  <div class="row">
3    <div class="col-4 offset-4">
4      {% if form.errors %}
5        <div class="row">
6          <div class="col">
7            <div class="alert alert-danger alert-dismissible fade show" role="alert">
8              <strong>Usuario y contraseña no coinciden</strong>
9              <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
10         </div>
11       </div>
12     {% endif %}
13   </div>
14   <form action="{% url 'login' %}" method="post">
15     {% csrf_token %}
16     <div class="mb-3">
17       <label for="username" class="form-label">Email<span class="text-danger">*</span></label>
18       <input type="text" class="form-control" id="username" name="username" required>
19       {% if form.username.errors %}
20         <div class="error-message">{{ form.username.errors.0 }}</div>
21       {% endif %}
22     </div>
23     <div class="mb-3">
24       <label for="password" class="form-label">Contraseña<span class="text-danger">*</span></label>
25       <input type="password" class="form-control" id="password" name="password" required>
26       {% if form.password.errors %}
27         <div class="error-message">{{ form.password.errors.0 }}</div>
28       {% endif %}
29     </div>
30     <button type="submit" class="btn btn-primary">Ingresar</button>
31   </form>
32   <p class="mt-3">Si no se ha registrado, puede hacerlo <a href="{% url 'register' %}">AQUÍ</a></p>
```

```
book > 🗂 views.py > ...
27     return render(request, 'inputbook.html', context)
28
29     Codeium: Refactor | Explain
30     class CustomLoginView(SuccessMessageMixin, LoginView):
31         success_message = "Sesion Iniciada Exitosamente"
32         template_name = 'registration/login.html'
33         redirect_authenticated_user = True
34
35         Codeium: Refactor | Explain
36         class CustomLogoutView(LogoutView):
37             next_page = reverse_lazy('login')
38             Codeium: Refactor | Explain | Generate Docstring | ×
39             def dispatch(self, request, *args, **kwargs):
40                 response = super().dispatch(request, *args, **kwargs)
41                 messages.add_message(request, messages.WARNING, "Sesion Cerrada Exitosamente")
42                 return response
```

```
book > 🗂 views.py > ...
3     from .forms import BookModelForm
4     from django.contrib.messages.views import SuccessMessageMixin    La importa
5     from django.contrib.auth.views import LoginView, LogoutView    La importaci
6     from django.contrib import messages    La importación "django.contrib" no se
7     from django.urls import reverse_lazy    La importación "django.urls" no se
8     from django.contrib.auth.decorators import login_required    La importación
9
9     Codeium: Refactor | Explain | Generate Docstring | ×
10    def index(request):
11        context = {}
12        return render(request, "index.html", context)
13
13     Codeium: Refactor | Explain | Generate Docstring | ×
14    def listbook(request):
15        libros = BookModel.objects.all()
16        context = {"libros": libros}
17        return render(request, "listbook.html", context)
18
18     Codeium: Refactor | Explain | Generate Docstring | ×
19     @login_required
20     def inputbook(request):
21         if request.method == "POST":
22             form = BookModelForm(request.POST)
23             if form.is_valid():
24                 form.save()
25                 return redirect("index")
26             else:
27                 form = BookModelForm()
28                 context = {"form": form}
29                 return render(request, "inputbook.html", context)
29
```

127.0.0.1:8000/accounts/login/?next=/inputbook/

Todos los marcadores

Site Web Django Inicio Agregar libros Libros Acerca de Contacto Login Registro

Ingrese a su cuenta:

Email*

Contraseña*

Ingresar

Si no se ha registrado, puede hacerlo [AQUÍ](#)

127.0.0.1:8000/inputbook/

Todos los marcadores

Site Web Django Inicio Agregar libros Libros Acerca de Contacto Hola, Salir

Título:

Autor:

Valoración: Valor entre 0 y 10000

Enviar

❖ DESCRIPCIÓN

Desarrollo de Sitio Web de Django.

PRODUCTOS

[Libros](#)

CONTACTO

⌂ Dirección, calle, País

✉ info@example.com

📞 +01 234 567 88

📠 +01 234 567 89