

Actividad de la Lección

En esta actividad, los estudiantes pondrán en práctica lo aprendido sobre manejo y validación de formularios en Flask. El enfoque será crear formularios requeridos para el proyecto Capstone, validarlos y proporcionar retroalimentación al usuario. Esto les ayudará a integrar formularios en sus aplicaciones de manera eficiente y profesional.

Actividad

1. Ejercicio práctico:

- Crear un formulario Flask con WTForms para registrar usuarios, utiliza el ejemplo provisto en clase profesor o el código que se encuentra en el repositorio de la clase (<https://github.com/javierdastas/comp2052>).
- Validar los siguientes campos:
- Nombre: Obligatorio, mínimo 3 caracteres.
- Correo: Obligatorio, formato de correo.
- Contraseña: Obligatoria, mínimo 6 caracteres.
- Proporcionar mensajes de error personalizados.

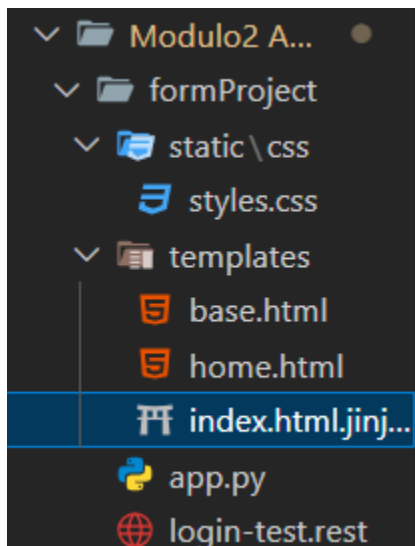
1. Diseño de Formularios para el Proyecto Capstone:

- Diseñar los formularios necesarios para el proyecto Capstone.
- Proporcionar una lista de validaciones necesarias para cada formulario.

1. Entrega:

- Subir el código del servidor Flask con tus formularios implementados a tu repositorio en Github.
- Desarrolla un documento en formato PDF que incluya:
- Capturas de pantallas de las pruebas realizadas a tu aplicación de Back-end.
- Incluye la dirección de tu Github que incluye copia de tu código.

Estructura basada en lo discutido en clase y lo visto en el GitHub del profesor:



Algunos de los estilos aplicados:

```
base.html home.html login-test.rest index.html.jinja2 app.py styles.css x
Modulo2 Apps > formProject > static > css > styles.css > .home-page
1  /* Estilos generales */
2  body {
3      background-color: #f4f4f4;
4      font-family: Arial, sans-serif;
5      margin: 0;
6      padding: 0;
7  }
8
9  /* Contenedor principal */
10 .container {
11     max-width: 800px; /* Ajuste el tamaño del contenedor para hacerlo más amplio */
12     margin: 40px auto;
13     background-color: white;
14     padding: 30px;
15     border-radius: 8px;
16     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
17 }
18
19 /* Título */
20 .title {
21     text-align: center;
22     margin-bottom: 20px;
23     font-weight: bold;
24     font-size: 2rem; /* Aumente el tamaño del título */
25 }
26
27 /* Estilo de formulario */
28 .process-form {
29     display: grid;
30     gap: 20px; /* Aumenté el espacio entre los elementos del formulario */
31     margin-bottom: 20px;
32 }
33
34 /* Estilos para los campos de texto y contraseñas */
35 .process-form input[type="text"],
36 .process-form input[type="password"] {
37     flex: 1;
```

Código en base.html:

```
base.html X home.html login-test.rest index.html.jinja2 app.py
Modulo2 Apps > formProject > templates > base.html > html > body > div.container > header
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="UTF-8" />
5     <title>{% block title %}Login - Mi Aplicación{% endblock %}</title>
6     <link rel="stylesheet" href="{{ url_for('static', filename='css/styles.css') }}">
7   </head>
8   <body>
9     <div class="container">
10      <header></header>
11      {% block content %}{% endblock %}
12      <footer></footer>
13    </div>
14  </body>
15 </html>
16
```

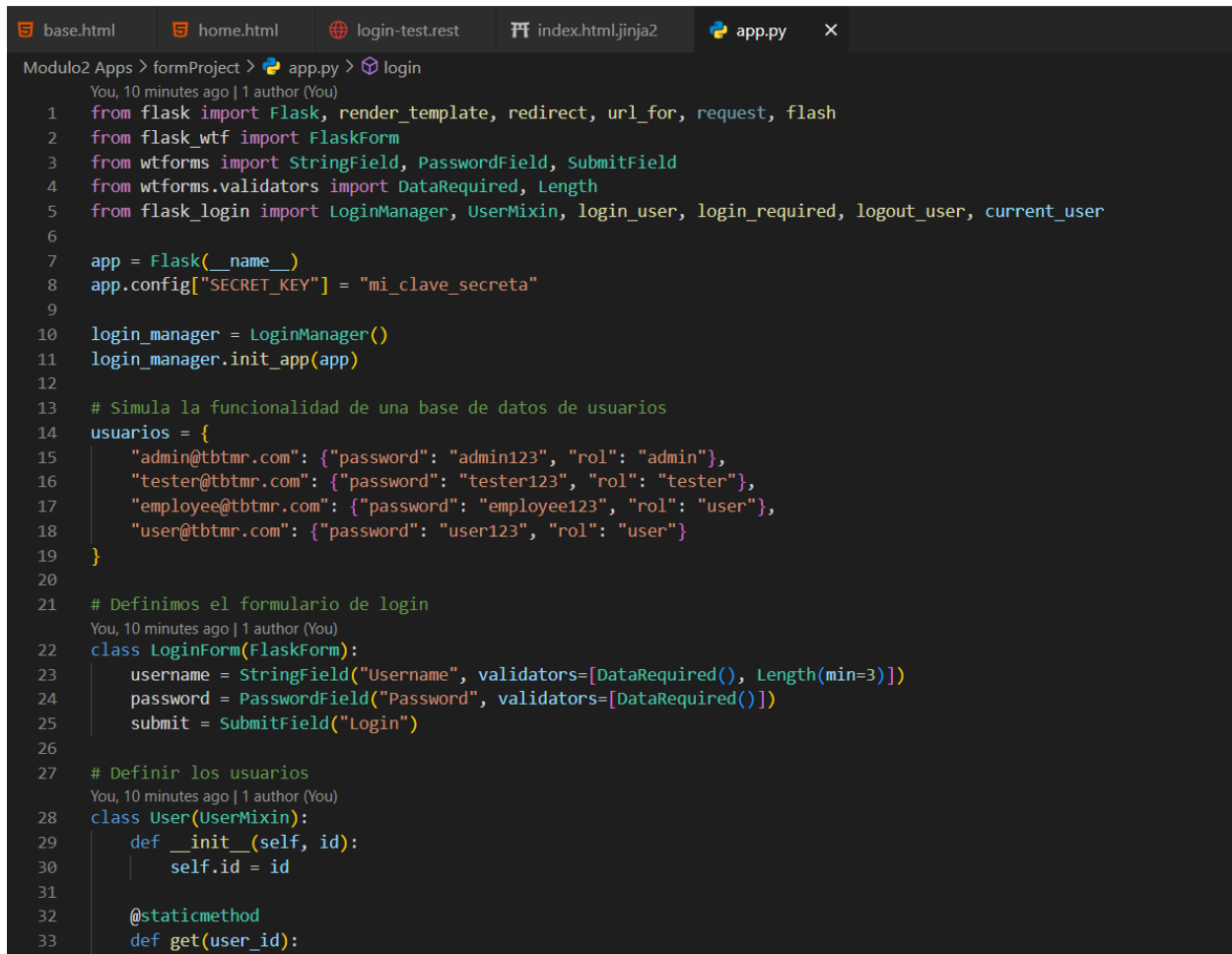
Código en home.html:

```
base.html home.html X login-test.rest index.html.jinja2
Modulo2 Apps > formProject > templates > home.html > ...
1 {% extends "base.html" %}
2
3 {% block content %}
4   <h1>Bienvenido, {{ message }}!</h1>
5   <p>Has iniciado sesión con éxito.</p>
6   <a href="{{ url_for('logout') }}">Cerrar sesión</a>
7 {% endblock %}
8
```

Código en index.html.jinja2:

```
base.html home.html login-test.rest index.html.jinja2 X app.py
Modulo2 Apps > formProject > templates > index.html.jinja2
1  {% extends "base.html" %}
2
3  {% block content %}
4      <h1 class="title">Login Form</h1>
5
6      <!-- Mostrar mensajes de flash (errores) -->
7      {% with messages = get_flashed_messages(with_categories=true) %}
8          {% if messages %}
9              <ul class="flashes">
10                 {% for category, message in messages %}
11                     <li class="{{ category }}">{{ message }}</li>
12                 {% endfor %}
13             </ul>
14         {% endif %}
15     {% endwith %}
16
17     <form method="POST" action="/login" class="process-form">
18         {{ form.hidden_tag() }}
19
20         <div>
21             <label for="username">Username:</label>
22             {{ form.username(class_="form-control") }}
23             {% for error in form.username.errors %}
24                 <p style="color: red;">{{ error }}</p>
25             {% endfor %}
26         </div>
27
28         <div>
29             <label for="password">Password:</label>
30             {{ form.password(class_="form-control") }}
31             {% for error in form.password.errors %}
32                 <p style="color: red;">{{ error }}</p>
33             {% endfor %}
34         </div>
35
36         <div>
37             {{ form.submit(class_="btn btn-primary") }}
38         </div>
39     </form>
40 {% endblock %}
41
```

Código del app.py:

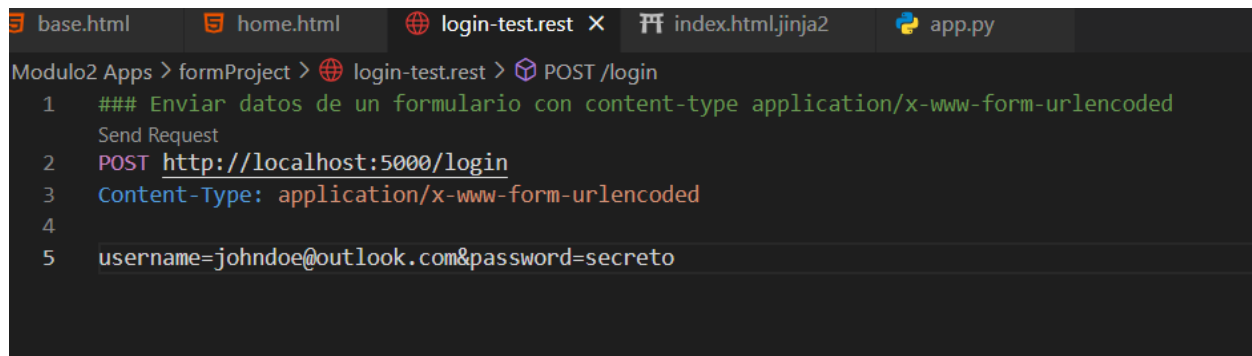


```
Modulo2 Apps > formProject > app.py > login
You, 10 minutes ago | 1 author (You)
1  from flask import Flask, render_template, redirect, url_for, request, flash
2  from flask_wtf import FlaskForm
3  from wtforms import StringField, PasswordField, SubmitField
4  from wtforms.validators import DataRequired, Length
5  from flask_login import LoginManager, UserMixin, login_user, login_required, logout_user, current_user
6
7  app = Flask(__name__)
8  app.config["SECRET_KEY"] = "mi_clave_secreta"
9
10 login_manager = LoginManager()
11 login_manager.init_app(app)
12
13 # Simula la funcionalidad de una base de datos de usuarios
14 usuarios = {
15     "admin@tbtmr.com": {"password": "admin123", "rol": "admin"},
16     "tester@tbtmr.com": {"password": "tester123", "rol": "tester"},
17     "employee@tbtmr.com": {"password": "employee123", "rol": "user"},
18     "user@tbtmr.com": {"password": "user123", "rol": "user"}
19 }
20
21 # Definimos el formulario de login
22 You, 10 minutes ago | 1 author (You)
23 class LoginForm(FlaskForm):
24     username = StringField("Username", validators=[DataRequired(), Length(min=3)])
25     password = PasswordField("Password", validators=[DataRequired()])
26     submit = SubmitField("Login")
27
28 # Definir los usuarios
29 You, 10 minutes ago | 1 author (You)
30 class User(UserMixin):
31     def __init__(self, id):
32         self.id = id
33
34     @staticmethod
35     def get(user_id):
```

```
34     return User(user_id)
35
36 @login_manager.user_loader
37 def load_user(user_id):
38     return User.get(user_id)
39
40 @app.route("/login", methods=["GET", "POST"])
41 def login():
42     form = LoginForm()
43     if form.validate_on_submit():
44         username = form.username.data
45         password = form.password.data
46
47         if username not in usuarios:
48             flash("El usuario no existe", "danger")
49         else:
50             usuario = usuarios[username]#envia un mensaje de error personalizado dependiendo del rol
51             if usuario["password"] != password:
52                 rol = usuario["rol"]
53                 flash(f"Contraseña incorrecta para el rol '{rol}'", "danger")
54             else:
55                 user = User(username)
56                 login_user(user)
57                 return redirect(url_for("home"))
58
59     return render_template("index.html.jinja2", form=form)
60
61
62 # Ruta de home.html
63 @app.route("/home")
64 @login_required
65 def home():
66     return render_template("home.html", message=current_user.id)
67
68 # Ruta para logout
69 @app.route("/logout")
```

```
70 @login_required
71 def logout():
72     logout_user()
73     return redirect(url_for("login"))
74
75 if __name__ == "__main__":
76     app.run(debug=True)
77
```

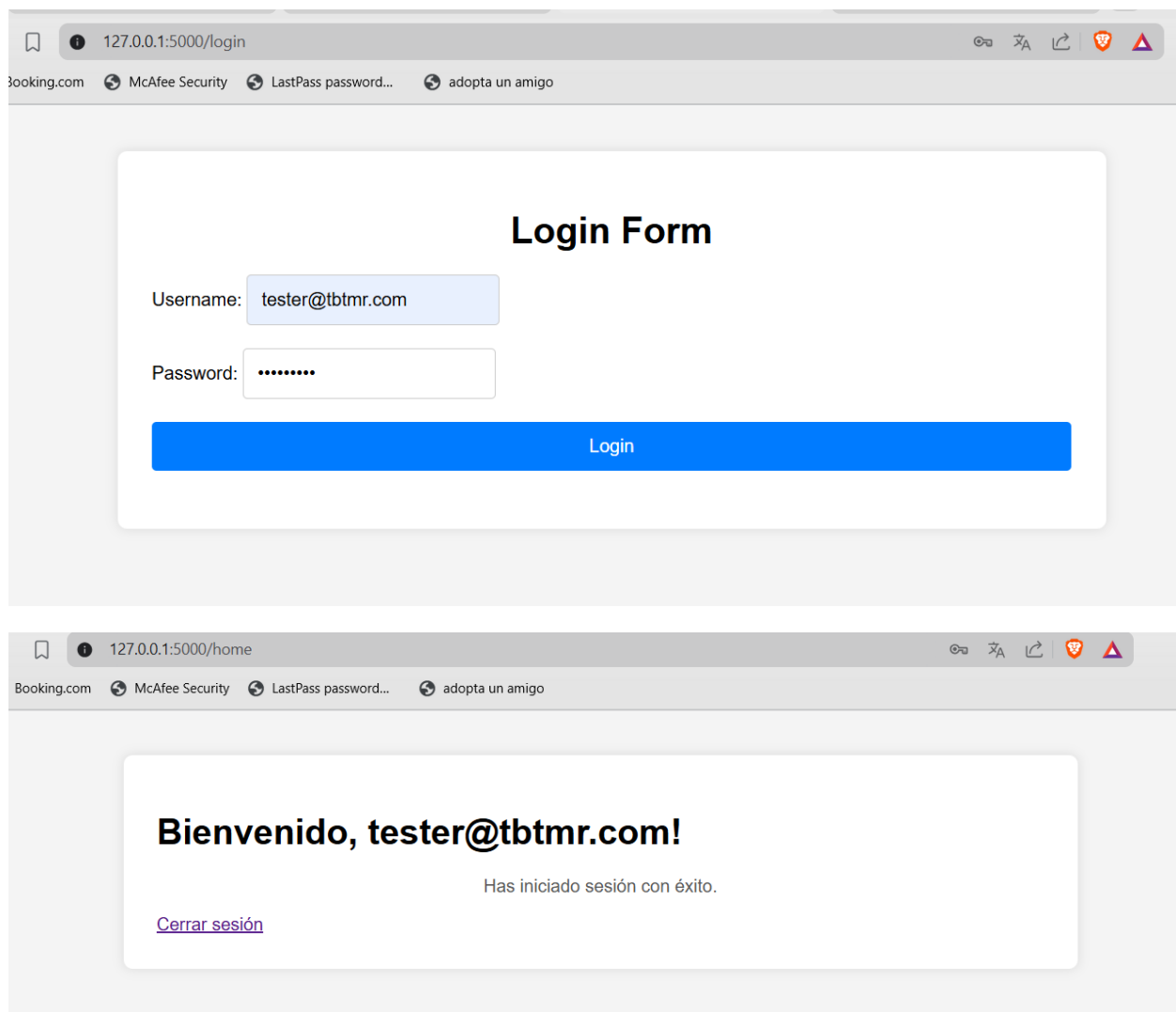
Código del client.rest:



```
base.html  home.html  login-test.rest X  index.html,jinja2  app.py

Modulo2 Apps > formProject > login-test.rest > POST /login
1  ### Enviar datos de un formulario con content-type application/x-www-form-urlencoded
   Send Request
2  POST http://localhost:5000/login
3  Content-Type: application/x-www-form-urlencoded
4
5  username=johndoe@outlook.com&password=secreto
```

Prueba del código en localhost:



127.0.0.1:5000/login

ig.com McAfee Security LastPass password... adopta un amigo

Login Form

- Contraseña incorrecta para el rol 'tester'

Username:

Password:

Login

127.0.0.1:5000/login

Booking.com McAfee Security LastPass password... adopta un amigo

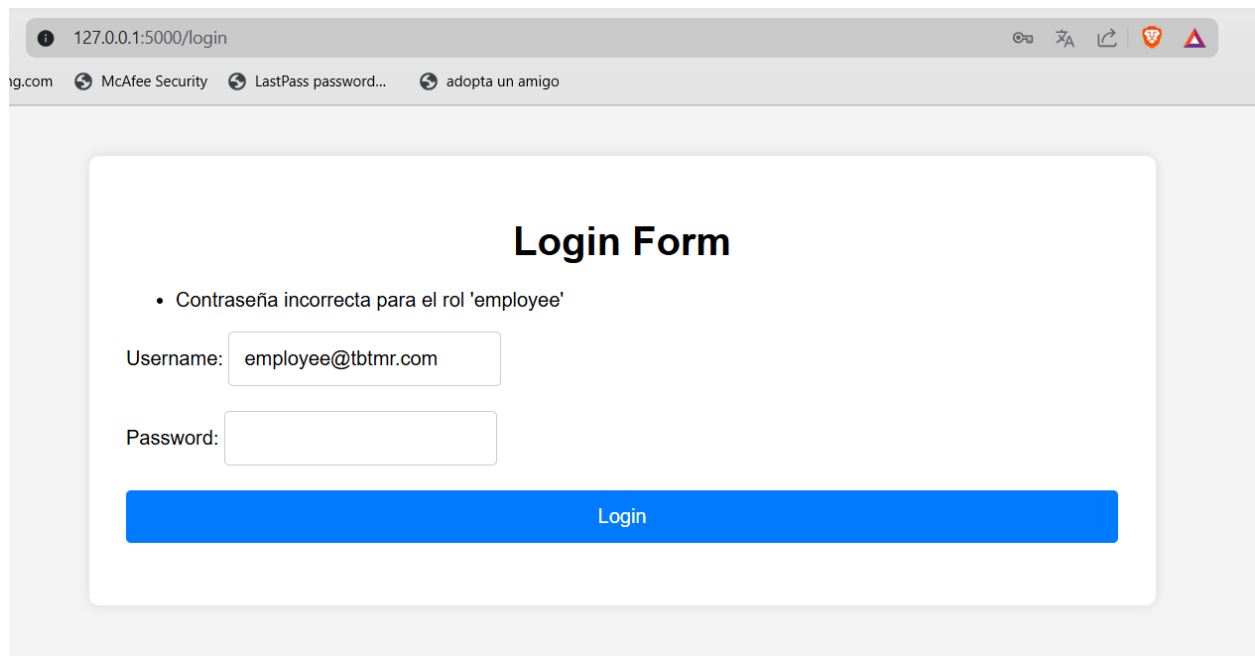
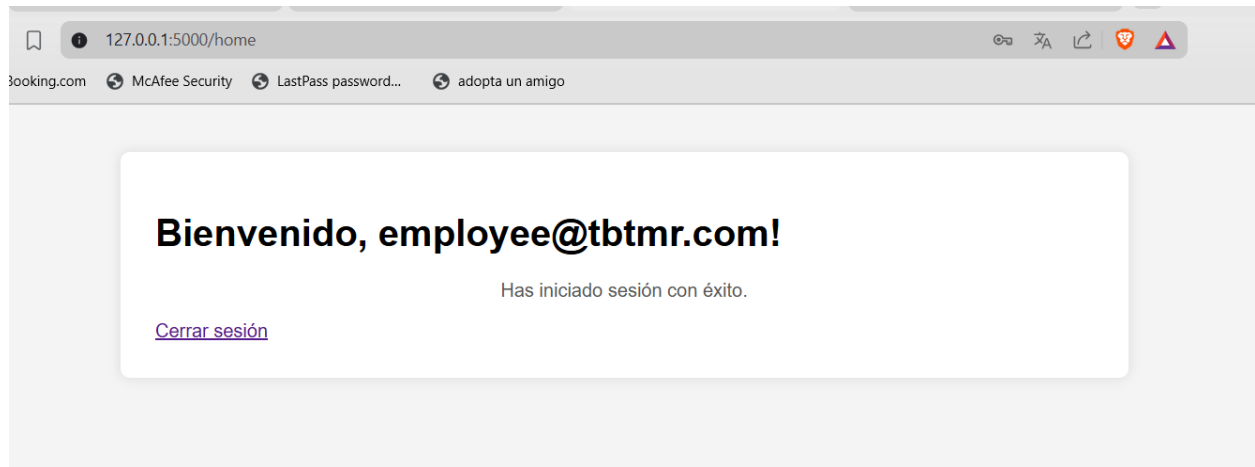
Login Form

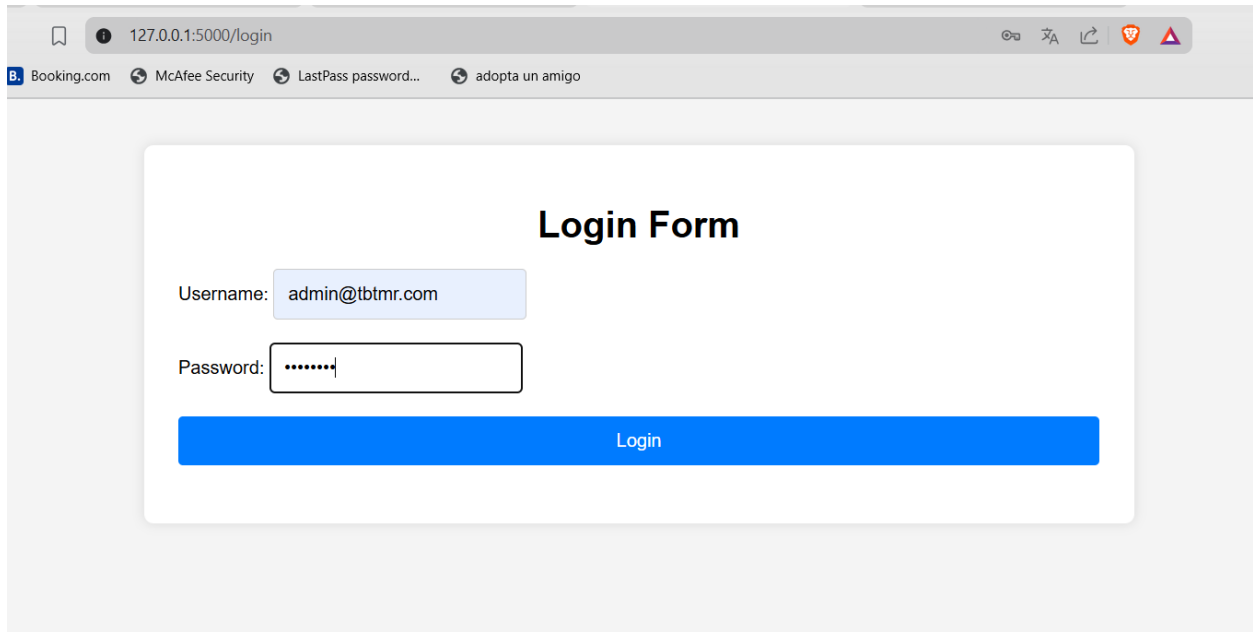
- El usuario no existe

Username:

Password:

Login





127.0.0.1:5000/login

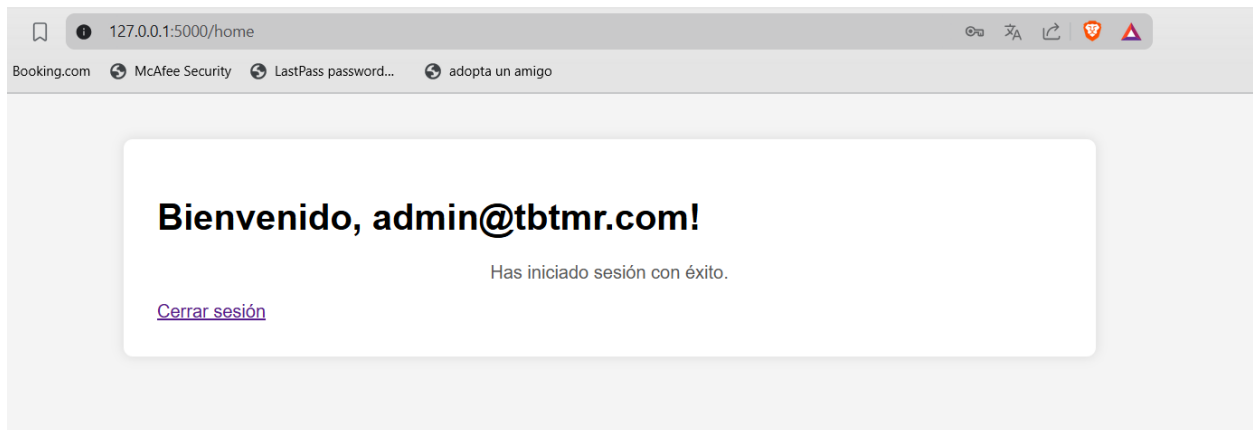
Booking.com McAfee Security LastPass password... adopta un amigo

Login Form

Username: admin@tbtmr.com

Password:

Login



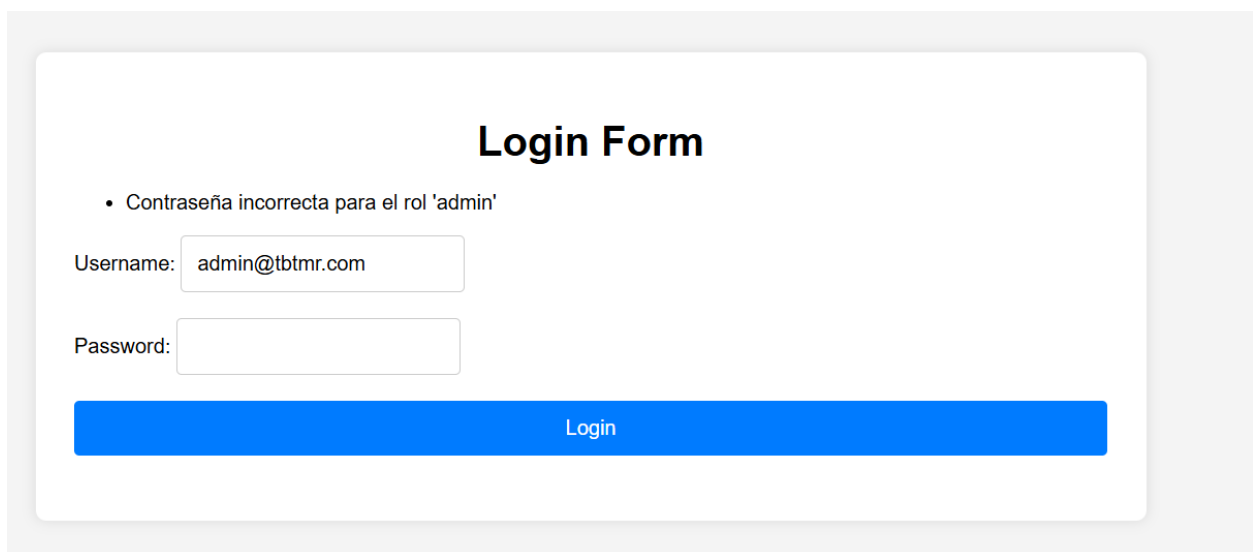
127.0.0.1:5000/home

Booking.com McAfee Security LastPass password... adopta un amigo

Bienvenido, admin@tbtmr.com!

Has iniciado sesión con éxito.

[Cerrar sesión](#)



127.0.0.1:5000/login

Booking.com McAfee Security LastPass password... adopta un amigo

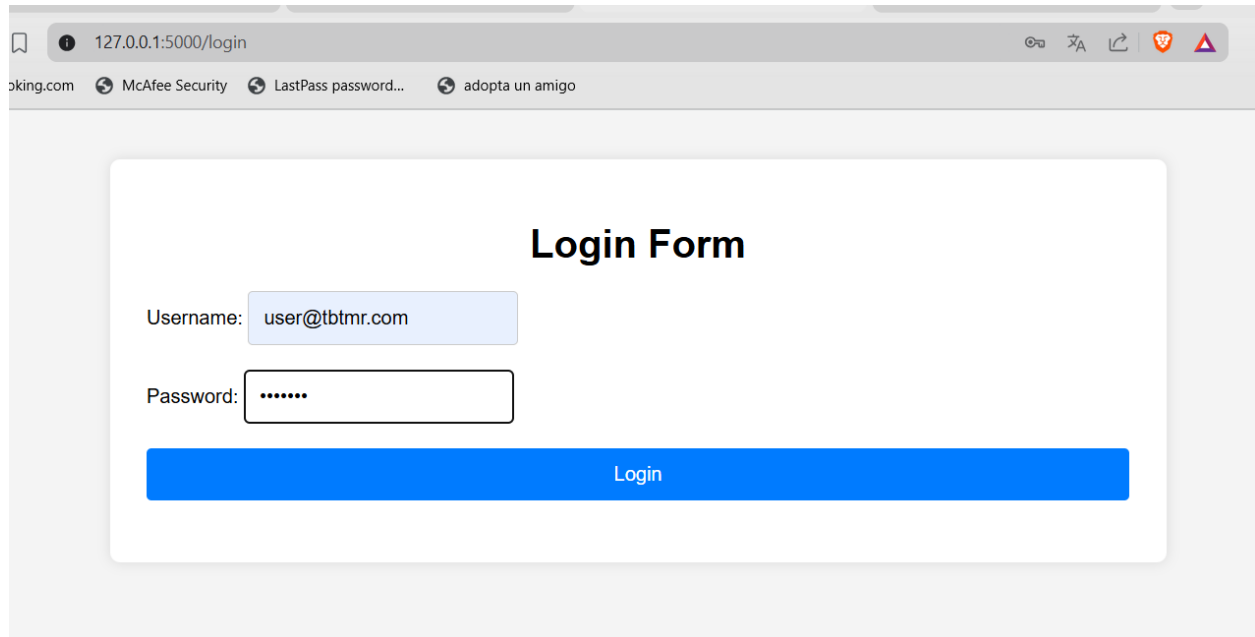
Login Form

- Contraseña incorrecta para el rol 'admin'

Username: admin@tbtmr.com

Password:

Login

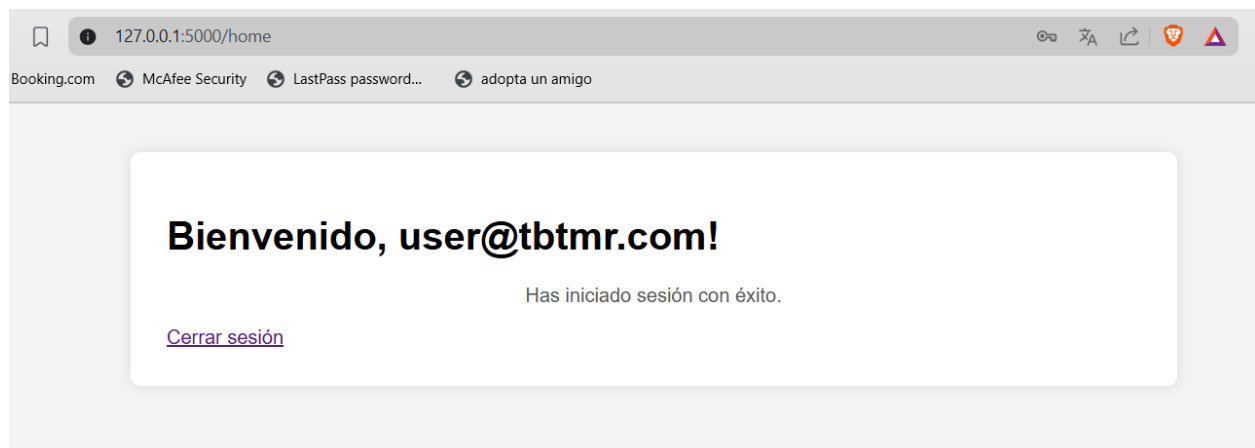


127.0.0.1:5000/login

Username: user@tbtmr.com

Password:

Login



127.0.0.1:5000/home

Bienvenido, user@tbtmr.com!

Has iniciado sesión con éxito.

[Cerrar sesión](#)

Login Form

- Contraseña incorrecta para el rol 'user'

Username:

Password:

Login

127.0.0.1:5000/login

McAfee Security LastPass password... adopta un amigo

Login Form

- El usuario no existe

Username:

Password:

Login

Link para el GitHub:

<https://github.com/Drewster64/microserver/tree/master/Modulo2%20Apps/formProject>