



El futuro digital
es de todos

MinTIC



CICLO III: Desarrollo de software

Mision
TIC2022





El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

Sesión 10: Desarrollo Software

Comunicación entre Front-end y Back-end

Mision
TIC2022



Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Manejo de formularios (continuación).
2. Intercambio de datos entre el servidor y el cliente.



Desarrollo Web

Front y Back: Flask – Formularios

Los formularios al permitir la interacción de los usuarios con la aplicación, se han convertido en uno de los elementos más básicos de las aplicaciones web. Para poder manejar formularios en Flask, se utiliza la extensión Flask-WTF la cual permite implementar el paquete WTForms en la aplicación junto con tokens CSRF, carga de archivos y ReCaptchas. Este paquete hace más sencilla la definición de formularios y el manejo de sus envíos.

Las extensiones de Flask son paquetes de Python regulares que pueden ser instalados con pip. Con el entorno virtual activo, se ejecuta el siguiente comando para instalar Flask-WTF:

```
(nombre_entorno) $ pip install flask-wtf
```



Desarrollo Web

Front y Back: Flask – Formularios

La extensión de Flask-WTF utiliza las clases de Python para representar los formularios. Una clase de formulario define los campos del formulario como variables de la clase.

- Crear un archivo con nombre forms.py en de la carpeta app, donde se guardaran los formularios.
- Formulario de inicio de sesión, el cual le solicitará al usuario su nombre de usuario y su contraseña.
- El formulario tendrá un botón para recordar las credenciales y un botón de inicio de sesión.



Desarrollo Web Front y Back: Flask – Formularios

Formulario de Inicio de Sesión

```
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, BooleanField, SubmitField
from wtforms.validators import DataRequired

class FormInicio(FlaskForm):
    usuario = StringField('Usuario', validators=[DataRequired(message='No dejar vacío, completar')])
    contraseña = PasswordField('Contraseña', validators=[DataRequired(message='No dejar vacío, completar')])
    recordar = BooleanField('Recordar Usuario')
    enviar = SubmitField('Iniciar Sesión')
```



Desarrollo Web Front y Back: Flask – Formularios

- Las cuatro clases (StringField, PasswordField, BooleanField, SubmitField) que representan los campos implementados en este formulario, son directamente importados desde WTForms, Flask-WTF no incluye versiones personalizadas.
- Se crea la referencia de cada campo en la clase FormInicio.
- Cada campo hereda una descripción o etiqueta como primer argumento.
- El argumento validators es opcional. Se utiliza para incluir validaciones en los campos.
- DataRequired verifica que el campo no esté vacío al momento de enviarlo.



Desarrollo Web Front y Back: Flask – Formularios

Plantillas de los Formularios

Para poder utilizar los formularios, se necesita que estén en su plantilla HTML específica, de forma que el explorador pueda renderizarla. En la carpeta de las plantillas, templates, se crea un archivo llamado `iniciar_sesion.html` y a continuación se escribe el código de la derecha:

```
{% extends "base.html" %}
{% block contenido %}
    <h1>Iniciar Sesión</h1>
    <form action="" method="post" novalidate>
        {{ form.hidden_tag() }}
        <p>
            {{ form.usuario.label }}<br>
            {{ form.usuario(size=30) }}<br>
        </p>
        <p>
            {{ form.contraseña.label }}<br>
            {{ form.contraseña(size=30) }}<br>
        </p>
        <p>{{ form.recordar() }} {{ form.recordar.label }}</p>
        <p>{{ form.enviar() }}</p>
    </form>
{% endblock %}
```




Desarrollo Web

Front y Back: Flask – Formularios

- Esta plantilla, espera recibir un formulario como argumento de la función view (FormInicio), el cual es referido como form.
- La etiqueta HTML <form> es utilizada de contenedor para el formulario.
- base.html es la plantilla base donde se indican las partes comunes de todas las páginas y los bloques que las otras plantillas pueden reescribir.
- El atributo action se usa para informar al explorador que dirección URL debe utilizar al momento de enviar el formulario lleno.
- El atributo method especifica qué método de solicitud HTTP debe ser utilizado al enviar el formulario al servidor.
- La expresión form.hidden_tag() de la plantilla, genera un campo oculto que incluye un token que se usa para proteger el formulario de ataques a través de CSRF.



Desarrollo Web Front y Back: Flask – Formularios

La vista del Formulario

Ahora hay que crear una función vista antes de poder ver los formularios en el explorador que se encargue del renderizado de la plantilla del formulario. Esta función estará en el archivo rutas.py.

```
from app import app
from flask import render_template
from app.forms import FormInicio
@app.route('/')
@app.route('/index')
def index():
    usuario = {'usuario':'...'}
    comentarios = [
        {
            'autor':{'usuario':'...'},
            'comentario':'...'
        },
        {
            'autor':{'usuario':'...'},
            'comentario':'...'
        }
    ]
    return render_template('index.html', titulo="Inicio", usuario=usuario,
comentarios=comentarios)
@app.route('/login')
def login():
    form = FormInicio()
    return render_template('iniciar_sesion.html',
        titulo='Iniciar Sesión', form=form)
```



Desarrollo Web Front y Back: Flask – Formularios

Para que sea más fácil, se
agrega un enlace de inicio
de sesión a base.html

```
<html>
  <head>
    {% if titulo %}
      <title>{{ titulo }} - ABC</title>
    {% else %}
      <title>ABC</title>
    {% endif %}
  </head>
  <body>
    <div>ABC:
      <a href="/index">Inicio</a>
      <a href="/login">Iniciar Sesión</a>
    </div>
    <hr>
    {% block contenido %}{% endblock %}
  </body>
</html>
```



Desarrollo Web Front y Back: Flask – Formularios

Al iniciar el servidor, se debería poder ver el formulario en la página de la aplicación de la siguiente forma:

ABC: [Inicio Iniciar Sesión](#)

Iniciar Sesión

Usuario

Contraseña

☐ Recordar Usuario

Iniciar Sesión



Desarrollo Web Front y Back: Flask – Formularios

Recibir Datos de un Formulario

- Al presionar el botón “Iniciar Sesión”, el explorador mostrará el mensaje: “Method Not Allowed” (“Método no permitido”). Debido a que la función de inicio de sesión no está completa. Solo puede mostrar el formulario en la página, más no tiene la parte lógica que permite procesar los datos enviados por los usuarios.
- Flask-WTF permite que este proceso sea sencillo.
- A continuación, se muestra la versión actualizada de la función vista que acepta y valida datos enviados por usuarios desde el archivo rutas.py.



Desarrollo Web

Front y Back: Flask – Formularios

```
from wtforms import StringField
from flask_wtf import FlaskForm
from app import app
from flask import Flask, render_template, request, redirect, url_for, flash
from app.forms import FormInicio
@app.route('/')
@app.route('/index')
def index():
    usuario = {'usuario':'... '}
    comentarios = [
        {
            'autor':{'...':'...'},
            'comentario':'...'
        },
        {
            'autor':{'usuario':'...'},
            'comentario':'...'
        }
    ]
    return render_template('index.html', titulo="Inicio",
        usuario=usuario, comentarios=comentarios)
```



Desarrollo Web

Front y Back: Flask – Formularios

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    form = FormInicio()
    if(form.validate_on_submit()):
        flash('Inicio de sesión solicitado por el usuario {}'.format(form.usuario.data, form.recordar.data))
        return redirect(url_for('gracias'))
    return render_template('iniciar_sesion.html', form=form)
@app.route('/gracias')
def gracias():
    return render_template('gracias.html')
```

El método `form.validate_on_submit()` hace todo el trabajo de procesamiento del formulario. Cuando el explorador envía la solicitud GET para recibir la página web con el formulario, este método devolverá False, para que en ese caso la función vaya directamente al final y renderiza la plantilla.



Desarrollo Web Front y Back: Flask – Formularios

El archivo gracias.html es muy simple y coloca dentro de la carpeta templates.

```
{% extends "base.html" %}
{% block contenido %}
    <h1>Gracias por iniciar</h1>
{% endblock %}
```

Al ejecutar la función flash(), Flask guarda el mensaje, estos mensajes no aparecen mágicamente en las páginas web. Las plantillas de la aplicación deben renderizar estos mensajes “flasheados” de forma tal, que funcione con la interfaz del sitio. Para que esto suceda se deben agregar estos mensaje en la plantilla base, para asegurar que todas las plantillas hereden esta funcionalidad.



Desarrollo Web Front y Back: Flask – Formularios

```
<html>
  <head>
    <meta charset="UTF-8">
    {% if titulo %}
    <title>{{ titulo }} - Blog</title>
    {% else %}
    <title>Blog</title>
    {% endif %}
  </head>
  <body>
    <div>Blog:
      <a href="/index">Inicio</a>
      <a href="/login">Iniciar Sesion</a>
    </div>
    <hr>
    {% with mensajes = get_flashed_messages() %}
    {% if mensajes %}
    <ul>
      {% for mensaje in mensajes %}
      <li>{{ mensaje }}</li>
      {% endfor %}
    </ul>
```

```
    {% endif %}
    {% endwith %}
    {% block contenido %}{% endblock %}
  </body>
</html>
```



Desarrollo Web Front y Back: Flask – Formularios

Este debería ser el resultado final al ejecutar la aplicación, ingresar a la misma y luego presionar enviar:

ABC : Inicio Iniciar Sesion

- Inicio de sesión solicitado por el usuario ... , recordar=False

Gracias por iniciar



Desarrollo Web

Front y Back: Flask – Formularios

La estructura del proyecto es:

```
flask1
├── .flaskenv
├── ABC.py
├── app
│   ├── forms.py
│   ├── rutas.py
│   ├── __init__.py
│   ├── settings
│   │   ├── config.py
│   │   └── __init__.py
│   └── templates
│       ├── base.html
│       ├── gracias.html
│       ├── index.html
│       └── iniciar_sesion.html
```



El futuro digital
es de todos

MinTIC



Vigilada Mineducación

Ejercicios de práctica





El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

¡GRACIAS
POR SER PARTE DE
ESTA EXPERIENCIA
DE APRENDIZAJE!



Misión
TIC 2022