

Actividad de la Lección

En esta actividad, utilizando Flask, implementarás roles y permisos en un proyecto pequeño y luego en tu proyecto capstone. Esto te permitirá aplicar los conceptos de autorización en un contexto práctico, protegiendo rutas y configurando permisos.

Pasos de la Actividad:

- **Configurar el Proyecto:**

- Define y crea una aplicación Flask con rutas básicas, utiliza el código de ejemplo discutido en clase o el código en el repositorio del curso (<https://github.com/javierdastas/comp2052>).

- Definir roles y permisos en el back-end creado por usted.

- **Implementar Autorización:**

- Usar Flask-Principal para proteger rutas según roles.

- **Probar Funcionalidad:**

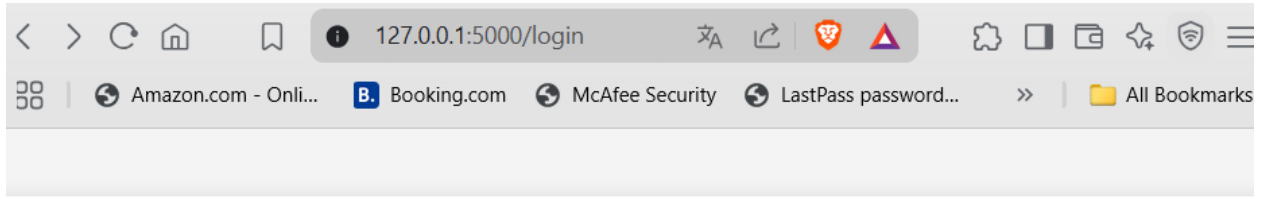
- Simular usuarios con diferentes roles y verificar los accesos permitidos y denegados.

1. Puedes usar VSCode (REST Client), Postman u otra alternativa para demostrar y documentar el funcionamiento de tu trabajo.

- **Entrega:**

- Subir el código del servidor Flask con la funcionalidad implementada a tu repositorio de GitHub.
- Desarrolla un documento en formato PDF que incluya:
 - Incluir un diagrama que muestre el flujo de autenticación del proyecto.
 - Captura de pantallas de las pruebas de la aplicación en Flask que desarrollaste.
- Incluye la dirección de tu repositorio en GitHub que incluye el código de esta actividad

Inicio de sesión como usuario regular:

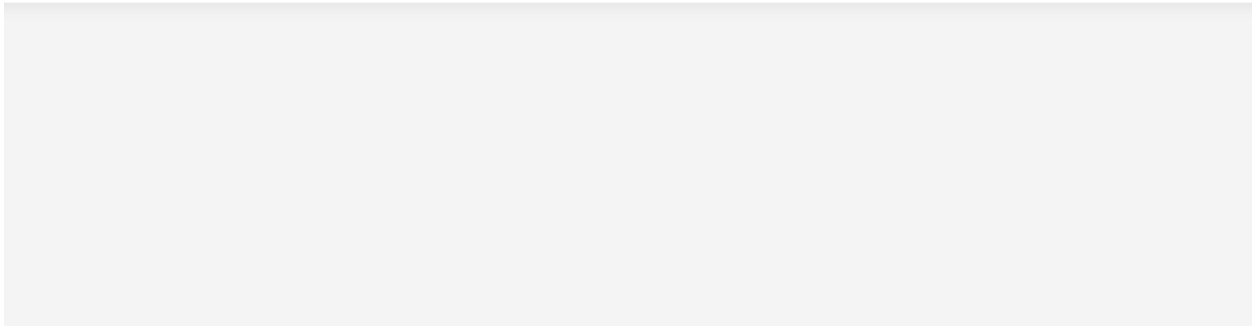


Login Form

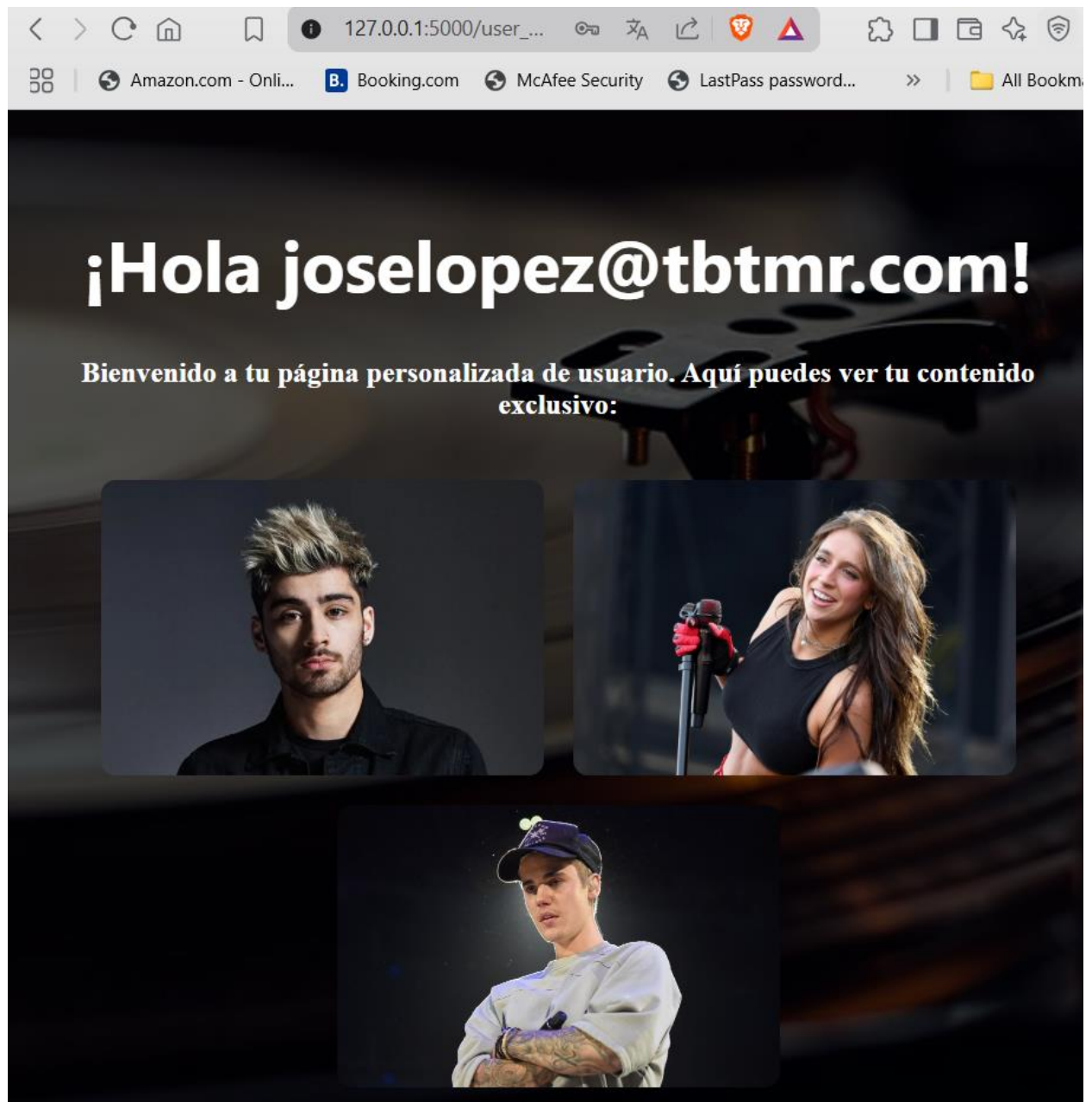
Username:

Password:

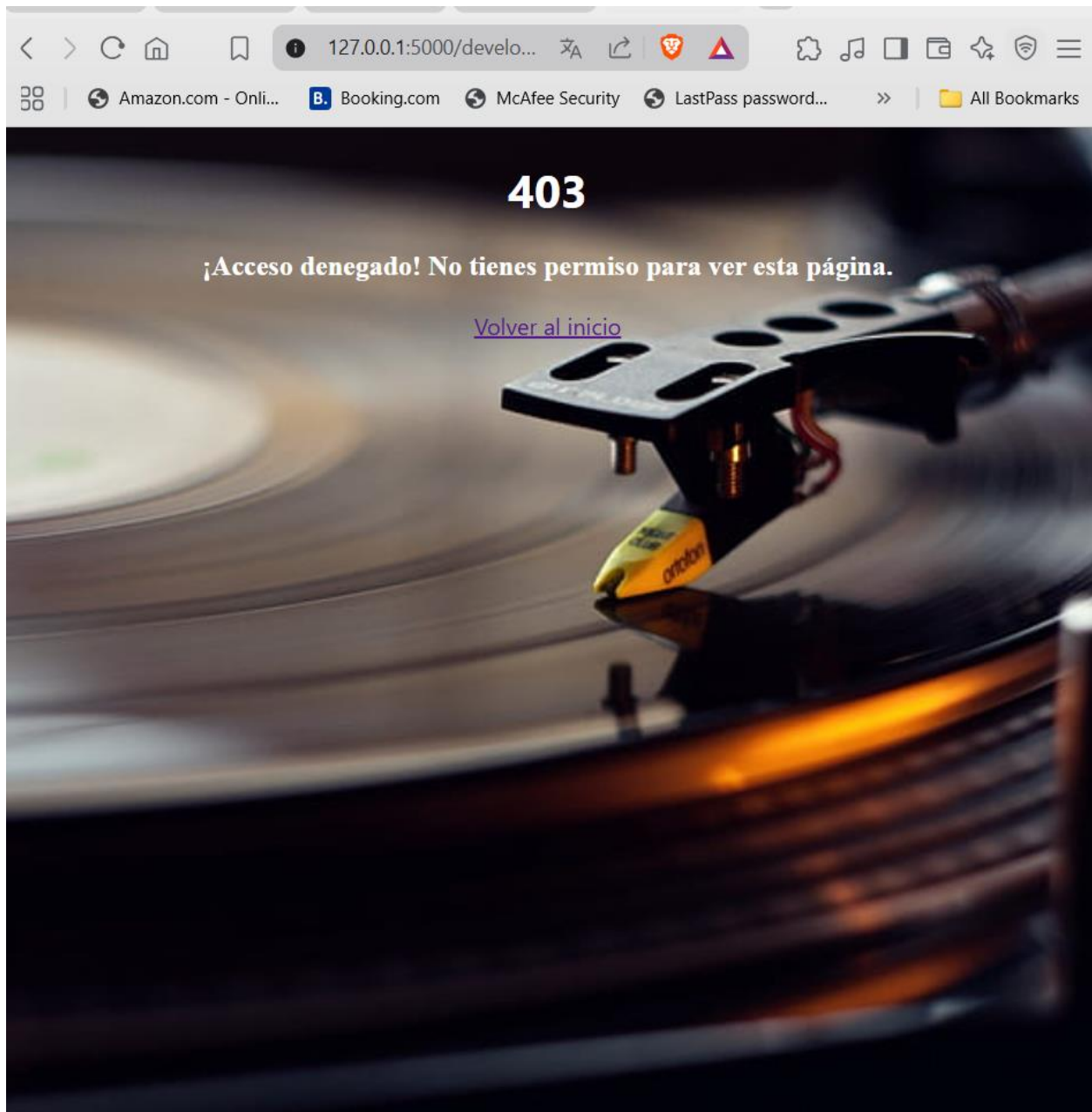
Login



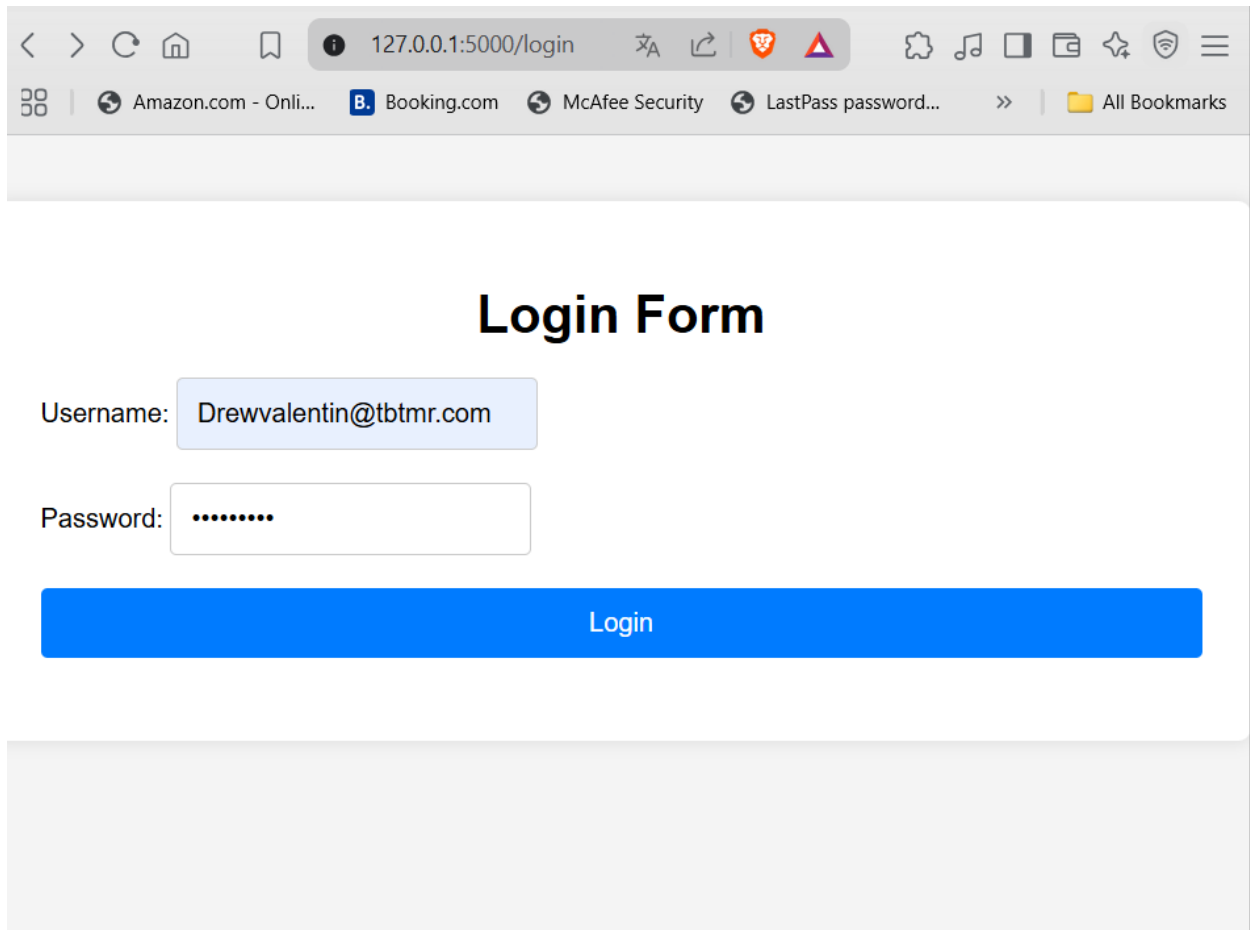
Página a la que tiene acceso un usuario regular:



Si el usuario inicia sesión como user (en este caso joselopez@tbtmr.com) y luego intenta acceder a la pagina de developer mediante el navegador (<http://127.0.0.1:5000/developer>), le da un mensaje informándole que no tiene acceso a este contenido:



Inicio de sesión como empleado:



The image shows a web browser window with a login form. The browser's address bar displays the URL "127.0.0.1:5000/login". The bookmarks bar includes "Amazon.com - Onli...", "Booking.com", "McAfee Security", "LastPass password...", and "All Bookmarks". The login form itself has a title "Login Form" in bold black text. Below the title, there are two input fields: "Username:" with the value "Drewvalentin@tbtmr.com" and "Password:" with masked characters ".....". A blue "Login" button is positioned below the password field.

127.0.0.1:5000/login

Amazon.com - Onli... Booking.com McAfee Security LastPass password... All Bookmarks

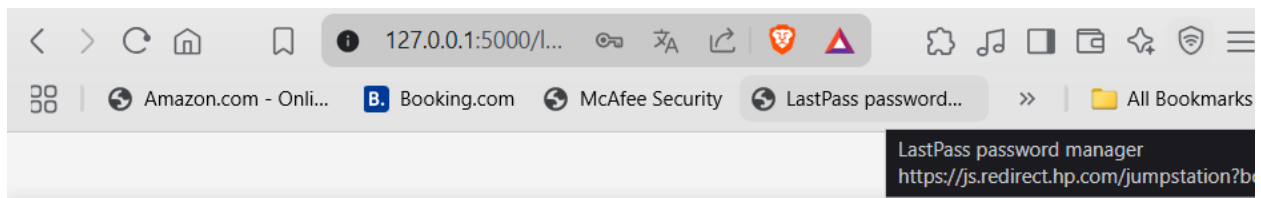
Login Form

Username: Drewvalentin@tbtmr.com

Password:

Login

Si la contraseña es incorrecta, la página lo informa:



Login Form

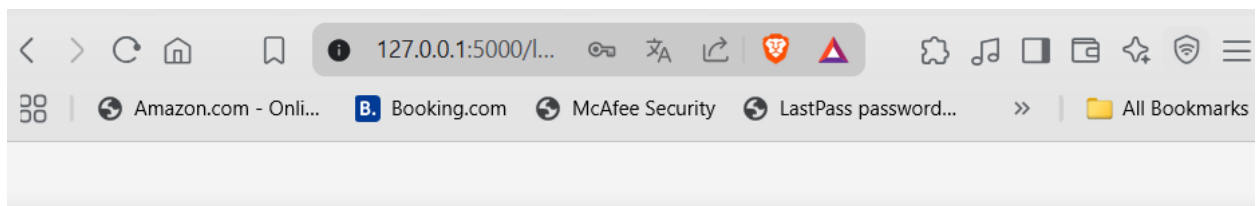
- Contraseña incorrecta para tester

Username:

Password:

Login

Si el usuario no existe, la página lo informa:



Login Form

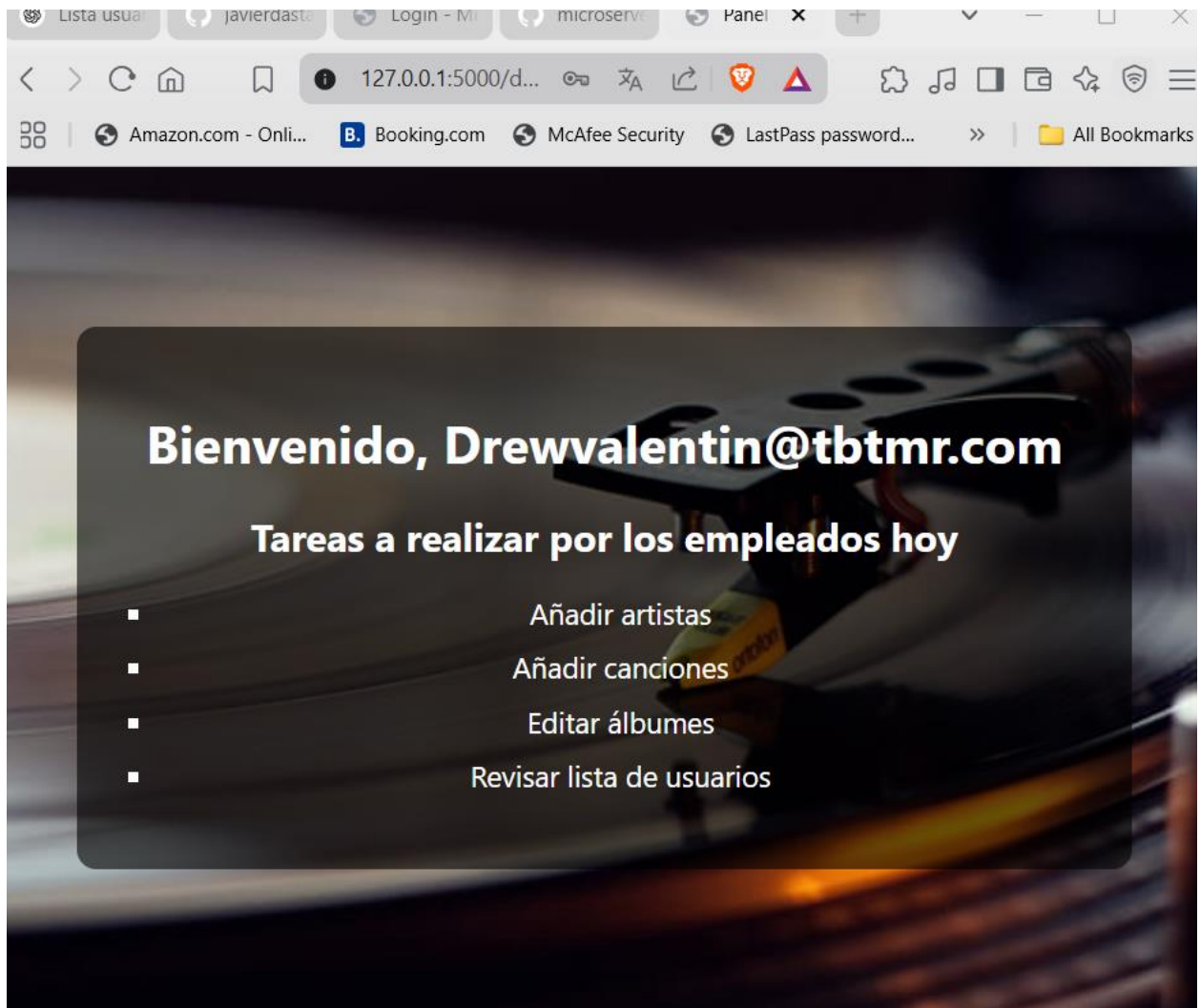
- Usuario no encontrado

Username:

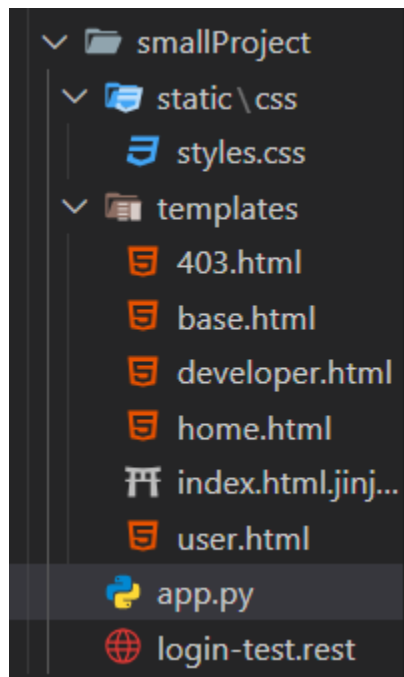
Password:

Login

Cuando se inicia sesión como alguno de los empleados establecidos, aparece esta pantalla mostrando tareas a realiza:



Estructura del proyecto:



Algunos de los estilos (están todos disponibles en GitHub):

```
Modulo2 Apps > smallProject > static > css > styles.css > tareas li
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"],
```

Código de 403.html:

```
app.py 403.html x developer.html user.html styles.css home.html
Modulo2 Apps > smallProject > templates > 403.html > html > body.user-page > p.letter
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Acceso Denegado</title>
6   <link rel="stylesheet" href="{{ url_for('static', filename='css/styles.css') }}">
7 </head>
8 <body class="user-page">
9   <h1>403</h1>
10  <p class="letter">¡Acceso denegado! No tienes permiso para ver esta página.</p>
11  <a href="{{ url_for('home') }}">Volver al inicio</a>
12 </body>
13 </html>
14
```

Código de base.html:

```
app.py 403.html base.html x developer.html user.html styles.css
Modulo2 Apps > smallProject > templates > base.html > ...
You, 17 hours ago | 1 author (You)
1 <!DOCTYPE html> You, 17 hours ago * modulo 3 leccion 2 -- version 1 (sin flask prin...
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 de developer.html:

```
app.py 403.html base.html developer.html X user.html styles.css
Modulo2 Apps > smallProject > templates > developer.html > html > body.user-page > div.content > h2
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Panel del Developer</title>
6   <link rel="stylesheet" href="{{ url_for('static', filename='css/styles.css') }}">
7 </head>
8 <body class="user-page">
9   <div class="content">
10     <h1>Bienvenido, {{ usuario }}</h1>
11     <h2>Tareas a realizar por los empleados hoy</h2>
12     <ul class="tareas">
13       <li>Añadir artistas</li>
14       <li>Añadir canciones</li>
15       <li>Editar álbumes</li>
16       <li>Revisar lista de usuarios</li>
17     </ul>
18   </div>
19 </body>
20 </html>
21
```

Código de home.html:

```
app.py 403.html developer.html home.html X user.html
Modulo2 Apps > smallProject > templates > home.html > ...
You, 17 hours ago | 1 author (You)
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
```

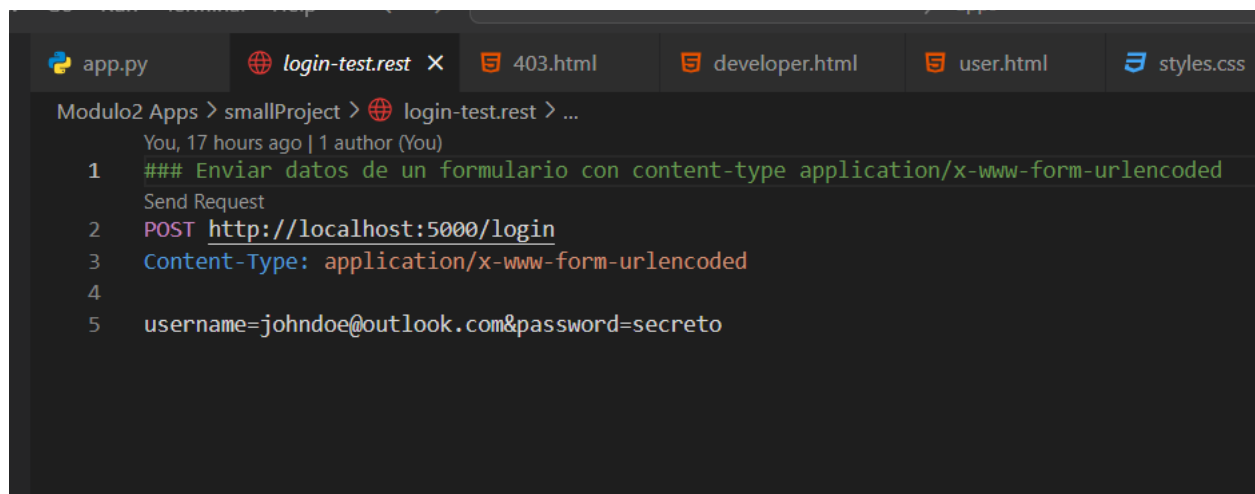
Código de user.html:

```
app.py 403.html developer.html index.html,jinja2 user.html x styles.css
Modulo2 Apps > smallProject > templates > user.html > html > body.user-page
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Bienvenido Usuario</title>
6      <link rel="stylesheet" href="{{ url_for('static', filename='css/styles.css') }}">
7  </head>
8  <body class="user-page">
9      <div class="overlay">
10         <h1>¡Hola {{ usuario }}!</h1>
11         <p class="letter">Bienvenido a tu página personalizada de usuario. Aquí puedes ver tu contenido exclusivo:</p>
12
13         <div class="gallery">
14             
15             
16             
17         </div>
18     </div>
19 </body>
20 </html>
21
```

Código de index.html.jinja:

```
app.py 403.html developer.html index.html.jinja2 x user.html styles.css
Modulo2 Apps > smallProject > templates > index.html.jinja2
You, 17 hours ago | 1 author (You)
1 {% extends "base.html" %} You, 17 hours ago * modulo 3 leccion 2 -- version 1 (sin flask prin...
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
```

Login.rest:



Código de app.py:

```
from flask import Flask, render_template, redirect, url_for, request, flash

from flask_wtf import FlaskForm

from wtforms import StringField, PasswordField, SubmitField

from wtforms.validators import DataRequired, Length

from flask_login import LoginManager, UserMixin, login_user, login_required, logout_user,
current_user

from flask_principal import Principal, Permission, RoleNeed, Identity, AnonymousIdentity,
identity_loaded, identity_changed


app = Flask(__name__)

app.config["SECRET_KEY"] = "mi_clave_secreta"


# Inicializar Flask-Login
login_manager = LoginManager()
login_manager.init_app(app)


# Inicializar Flask-Principal
principals = Principal(app)


# Base de datos simulada de usuarios
usuarios = {
    "emafranco@tbtml.com": {"password": "admin123", "rol": "admin"},
    "Drewvalentin@tbtml.com": {"password": "tester123", "rol": "tester"},
    "benmartinez@tbtml.com": {"password": "employee123", "rol": "employee"},
    "joselopez@tbtml.com": {"password": "user123", "rol": "user"}
}
```



```
# Permisos por rol
```

```
admin_permission = Permission(RoleNeed("admin"))
```

```
tester_permission = Permission(RoleNeed("tester"))
```

```
employee_permission = Permission(RoleNeed("employee"))
```

```
user_permission = Permission(RoleNeed("user"))
```

```
# Permiso combinado para developer
```

```
developer_permission = Permission(RoleNeed("admin"), RoleNeed("tester"),  
RoleNeed("employee"))
```

```
# Formulario de Login
```

```
class LoginForm(FlaskForm):
```

```
    username = StringField("Username", validators=[DataRequired(), Length(min=3)])
```

```
    password = PasswordField("Password", validators=[DataRequired()])
```

```
    submit = SubmitField("Login")
```

```
# Clase User para Flask-Login
```

```
class User(UserMixin):
```

```
    def __init__(self, id):
```

```
        self.id = id
```

```
    @staticmethod
```

```
    def get(user_id):
```

```
        return User(user_id)
```

```
@login_manager.user_loader
```

```
def load_user(user_id):
```

```

return User.get(user_id)

# Cargar identidad y asignar roles
@identity_loaded.connect_via(app)
def on_identity_loaded(sender, identity):
    identity.user = current_user

    if current_user.is_authenticated:
        user_id = current_user.get_id()
        rol = usuarios[user_id]["rol"]
        identity.provides.add(RoleNeed(rol))

# Ruta de login
@app.route("/login", methods=["GET", "POST"])
def login():
    form = LoginForm()

    if form.validate_on_submit():
        username = form.username.data
        password = form.password.data

        if username in usuarios:
            user_data = usuarios[username]

            if user_data["password"] == password:
                user = User(username)
                login_user(user)
                identity_changed.send(app, identity=Identity(user.id))

            rol = user_data["rol"]

```

```

        if rol in ["admin", "tester", "employee"]:
            return redirect(url_for("developer"))
        elif rol == "user":
            return redirect(url_for("user_page"))
        else:
            flash(f'Contraseña incorrecta para {user_data[rol]}', "danger")
        else:
            flash("Usuario no encontrado", "danger")

    return render_template("index.html.jinja2", form=form)

# Ruta de logout
@app.route("/logout")
@login_required
def logout():
    logout_user()
    identity_changed.send(app, identity=AnonymousIdentity())
    return redirect(url_for("login"))

# Página principal
@app.route("/home")
@login_required
def home():
    return render_template("home.html", message=current_user.id)

# Página protegida para admin/tester/employee
@app.route("/developer")

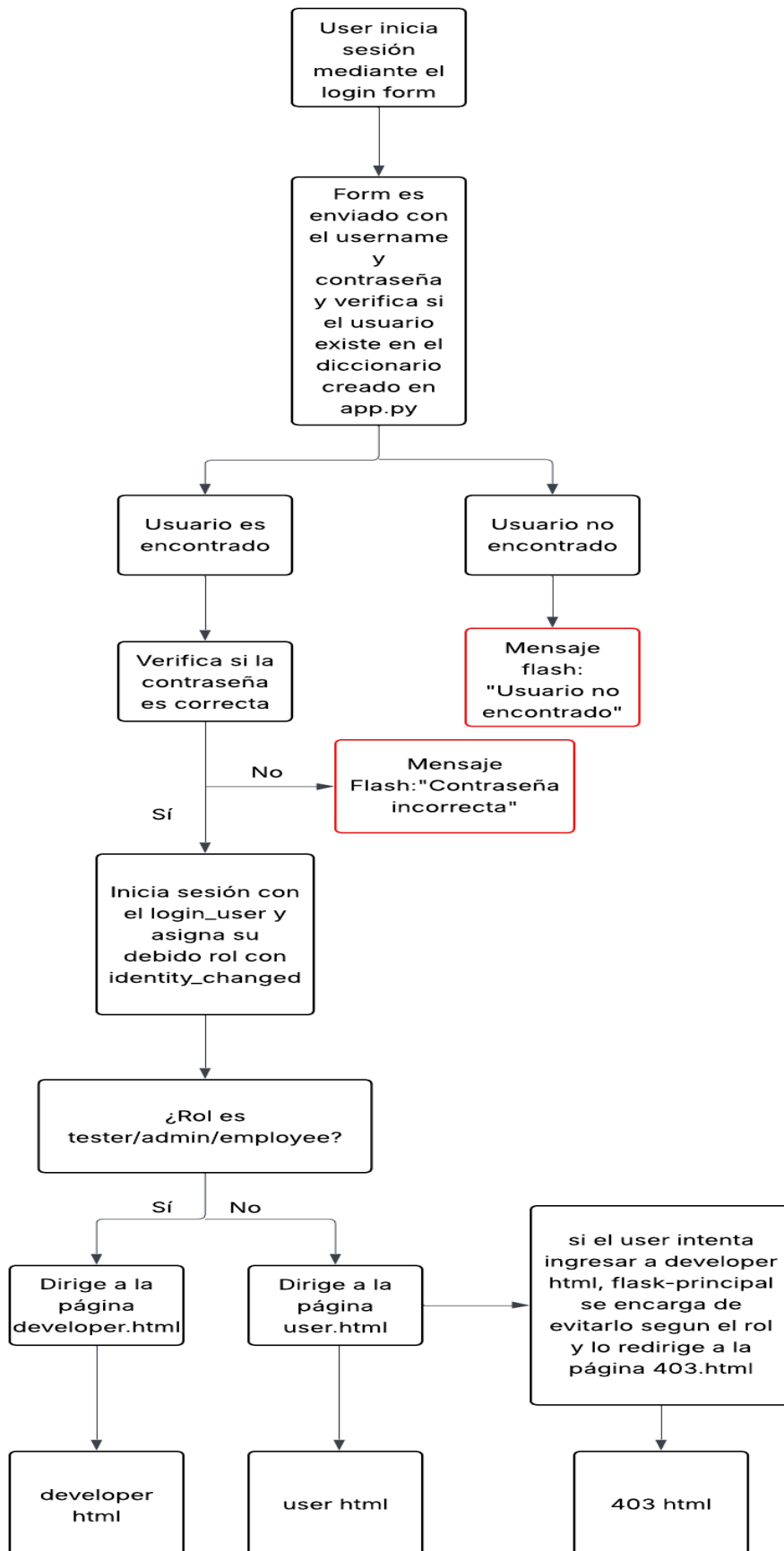
```

```
@login_required
@developer_permission.require(http_exception=403)
def developer():
    return render_template("developer.html", usuario=current_user.id)

# Página exclusiva para user
@app.route("/user_page")
@login_required
@user_permission.require(http_exception=403)
def user_page():
    return render_template("user.html", usuario=current_user.id)

# Página de error 403
@app.errorhandler(403)
def access_denied(e):
    return render_template("403.html"), 403

# Ejecutar la app
if __name__ == "__main__":
    app.run(debug=True)
```



Link al GitHub:

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