ESPIRITU HIELO

Integrantes:
Alvaro Aguirre 33%
Eduardo Salas 33%
Francesco Uccelli 33%

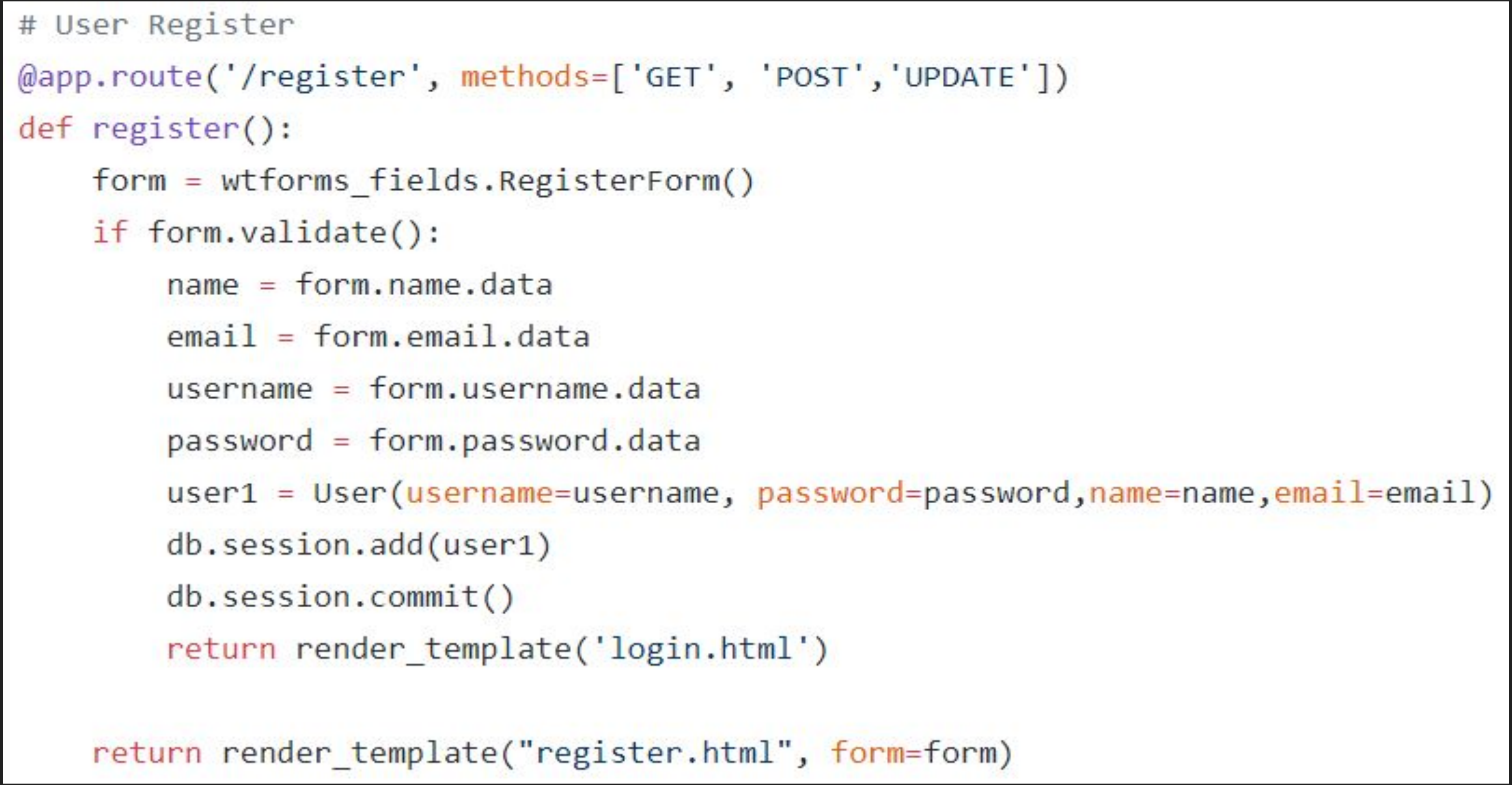
PROBLEMÁTICA

- Registro de pedidos semanales a mano
- Nada está automatizado
- Impresión de papeles innecesaria

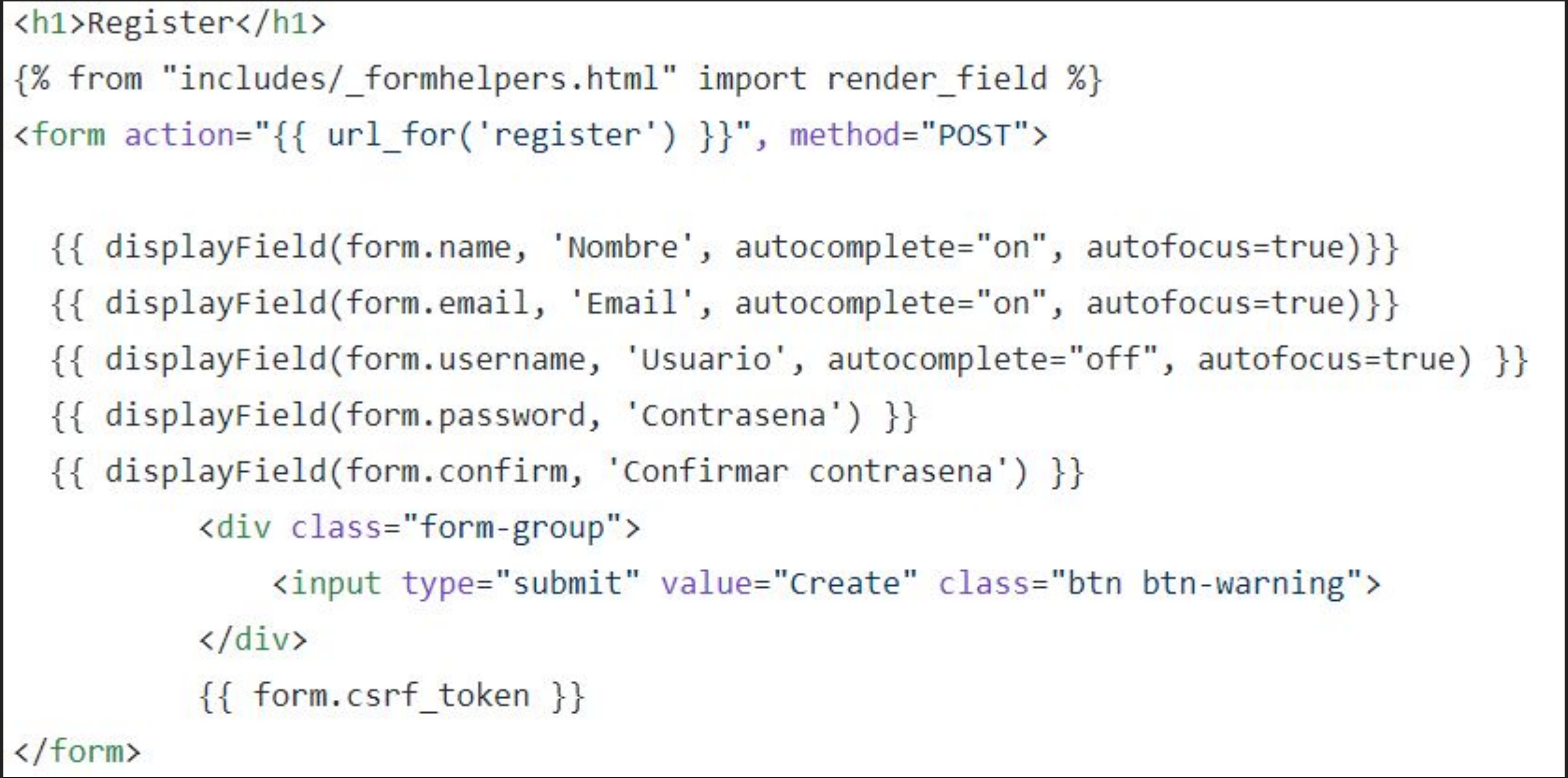
UTILIDAD

- Registra pedidos
- Fecha de entrega
- Almacena datos sobre los clientes

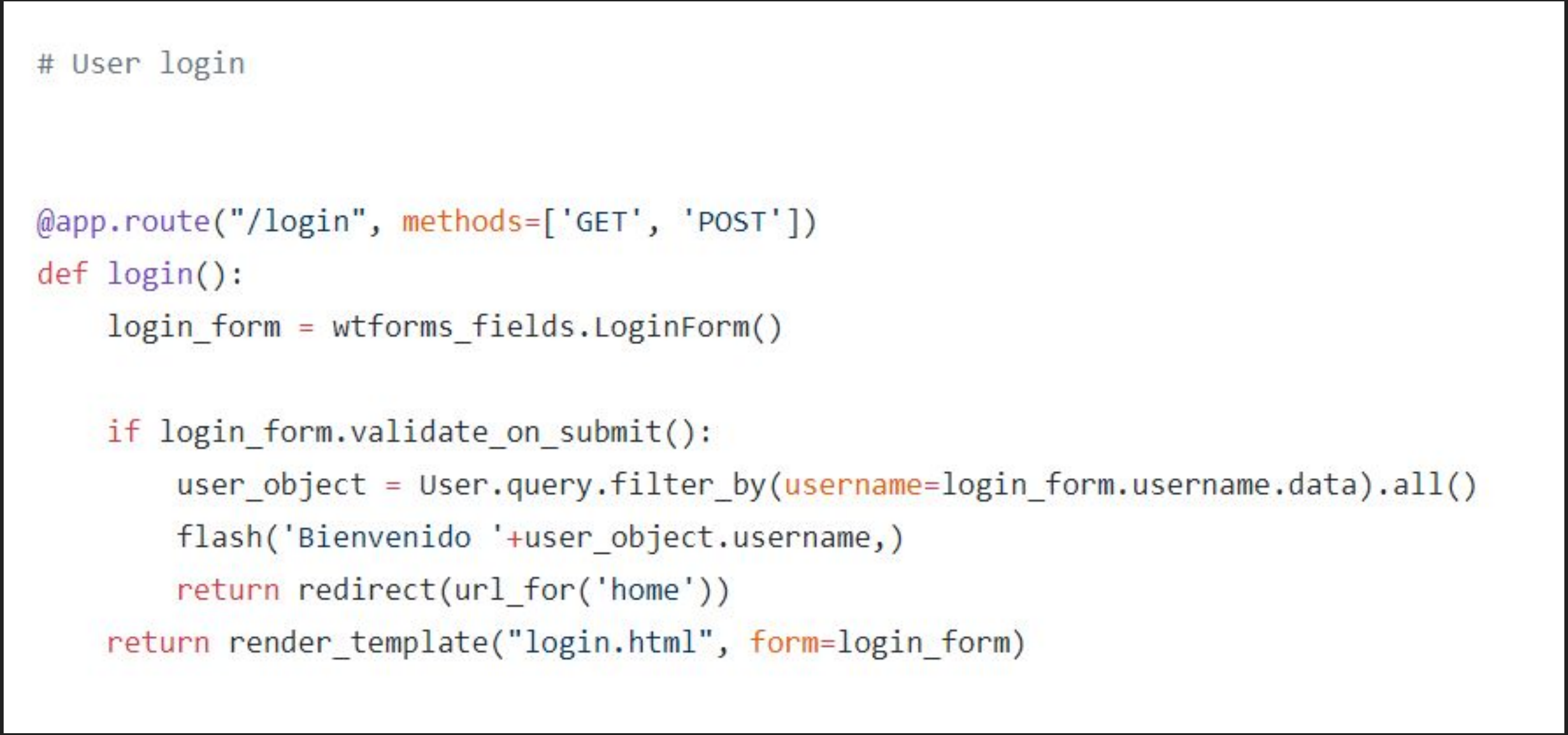
REGISTRO



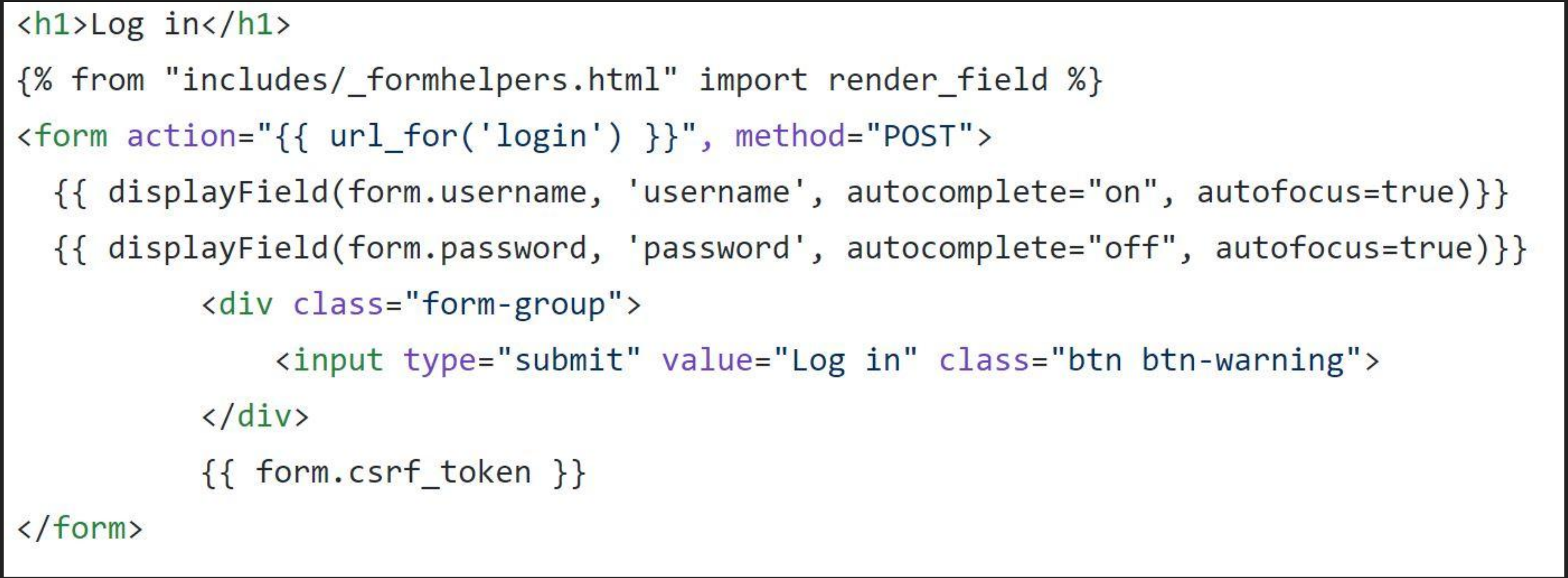
```
# User Register
@app.route('/register', methods=['GET', 'POST','UPDATE'])
def register():
    form = wtforms_fields.RegisterForm()
    if form.validate():
        name = form.name.data
        email = form.email.data
        username = form.username.data
        password = form.password.data
        user1 = User(username=username, password=password,name=name,email=email)
        db.session.add(user1)
        db.session.commit()
        return render_template('login.html')

    return render_template("register.html", form=form)
```


```
<h1>Register</h1>
{% from "includes/_formhelpers.html" import render_field %}
<form action="{{ url_for('register') }}", method="POST">

  {{ displayField(form.name, 'Nombre', autocomplete="on", autofocus=true)}}
  {{ displayField(form.email, 'Email', autocomplete="on", autofocus=true)}}
  {{ displayField(form.username, 'Usuario', autocomplete="off", autofocus=true) }}
  {{ displayField(form.password, 'Contrasena') }}
  {{ displayField(form.confirm, 'Confirmar contrasena') }}
          <div class="form-group">
              <input type="submit" value="Create" class="btn btn-warning">
          </div>
          {{ form.csrf_token }}
</form>
```

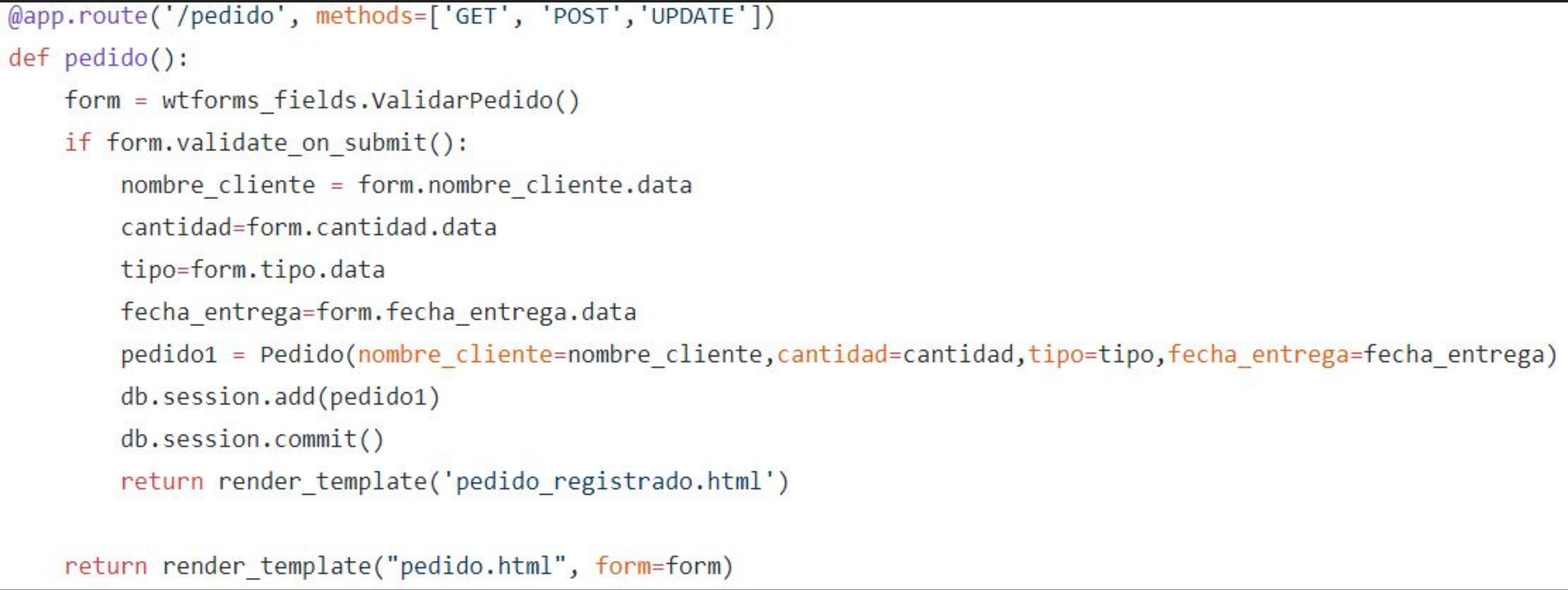
LOGIN



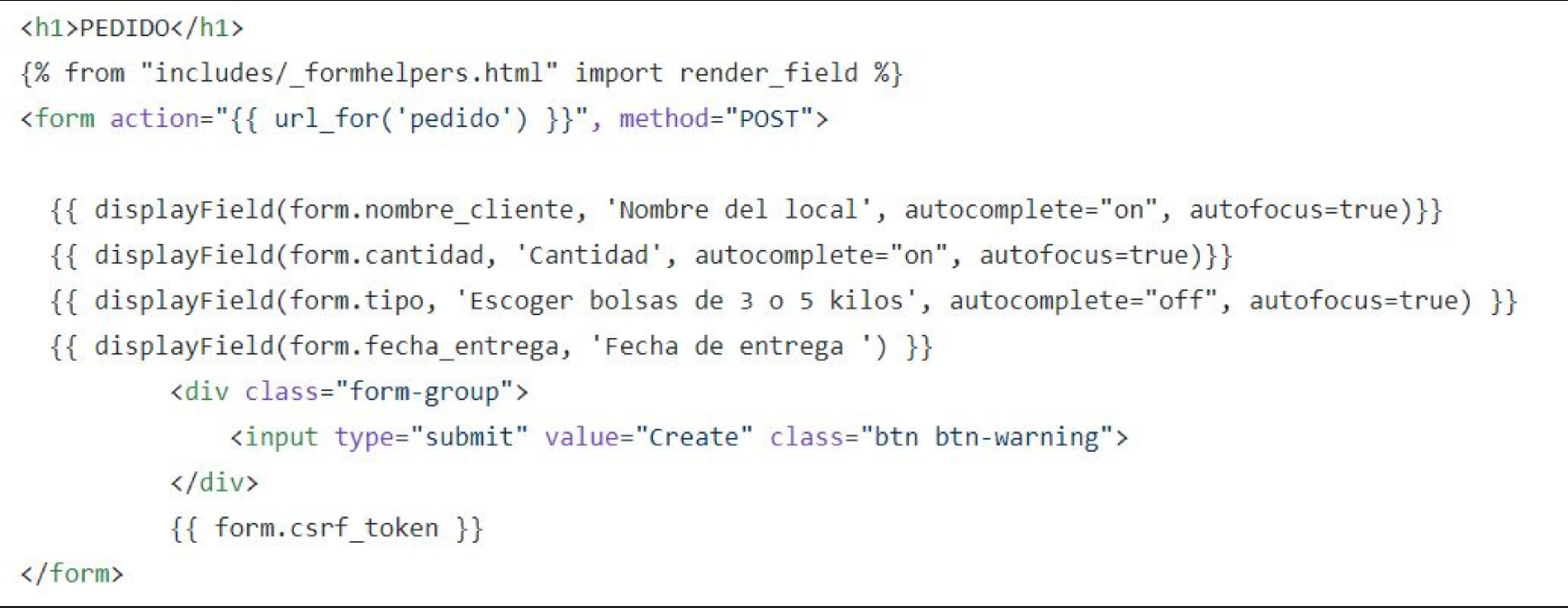
```
# User login



@app.route("/login", methods=['GET', 'POST'])
def login():
    login_form = wtforms_fields.LoginForm()

    if login_form.validate_on_submit():
        user_object = User.query.filter_by(username=login_form.username.data).all()
        flash('Bienvenido '+user_object.username,)
        return redirect(url_for('home'))
    return render_template("login.html", form=login_form)
```


```
<h1>Log in</h1>
{% from "includes/_formhelpers.html" import render_field %}
<form action="{{ url_for('login') }}", method="POST">
  {{ displayField(form.username, 'username', autocomplete="on", autofocus=true)}}
  {{ displayField(form.password, 'password', autocomplete="off", autofocus=true)}}
          <div class="form-group">
              <input type="submit" value="Log in" class="btn btn-warning">
          </div>
          {{ form.csrf_token }}
</form>
```

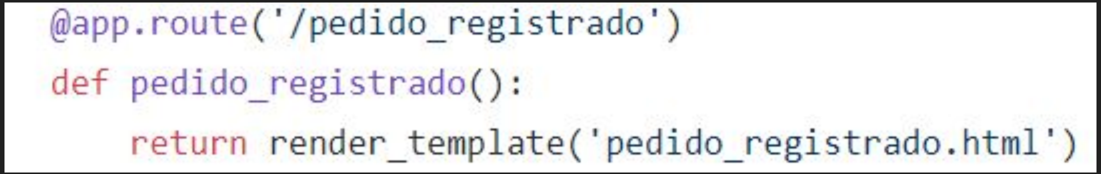
PEDIDO



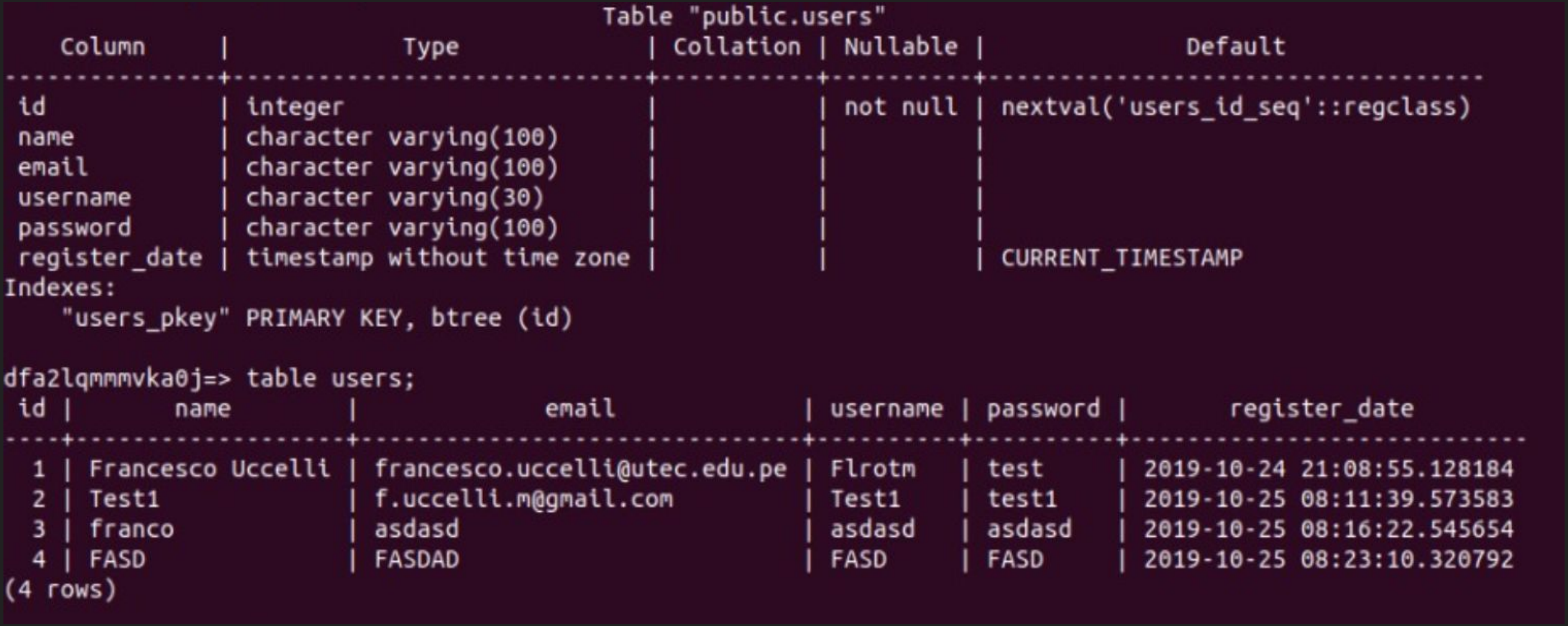
```
@app.route('/pedido', methods=['GET', 'POST','UPDATE'])
def pedido():
    form = wtforms_fields.ValidarPedido()
    if form.validate_on_submit():
        nombre_cliente = form.nombre_cliente.data
        cantidad=form.cantidad.data
        tipo=form.tipo.data
        fecha_entrega=form.fecha_entrega.data
        pedido1 = Pedido(nombre_cliente=nombre_cliente,cantidad=cantidad,tipo=tipo,fecha_entrega=fecha_entrega)
        db.session.add(pedido1)
        db.session.commit()
        return render_template('pedido_registrado.html')

    return render_template("pedido.html", form=form)
```



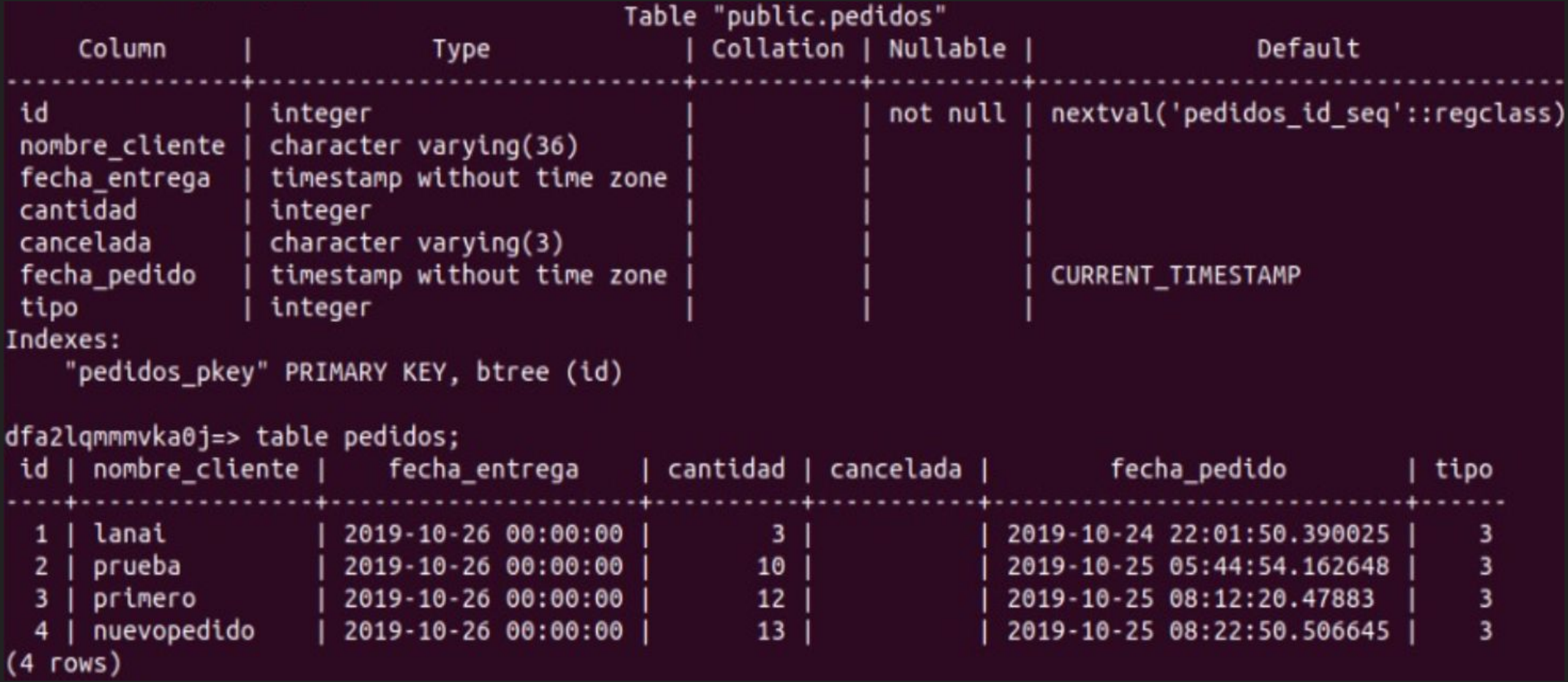
```
<h1>PEDIDO</h1>
{% from "includes/_formhelpers.html" import render_field %}
<form action="{{ url_for('pedido') }}", method="POST">

  {{ displayField(form.nombre_cliente, 'Nombre del local', autocomplete="on", autofocus=true)}}
  {{ displayField(form.cantidad, 'Cantidad', autocomplete="on", autofocus=true)}}
  {{ displayField(form.tipo, 'Escoger bolsas de 3 o 5 kilos', autocomplete="off", autofocus=true) }}
  {{ displayField(form.fecha_entrega, 'Fecha de entrega ') }}
          <div class="form-group">
              <input type="submit" value="Create" class="btn btn-warning">
          </div>
          {{ form.csrf_token }}
</form>
```



```python
@app.route('/pedido_registrado')
def pedido_registrado():
    return render_template('pedido_registrado.html')
```

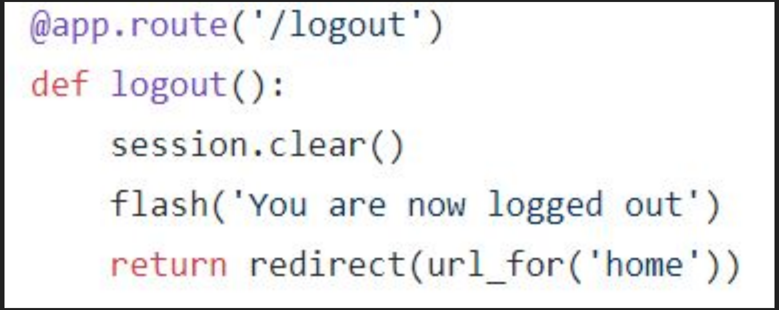


```
                              Table "public.users"
    Column     |            Type             | Collation | Nullable |              Default
---------------+-----------------------------+-----------+----------+-----------------------------------
 id            | integer                     |           | not null | nextval('users_id_seq'::regclass)
 name          | character varying(100)      |           |          |
 email         | character varying(100)      |           |          |
 username      | character varying(30)       |           |          |
 password      | character varying(100)      |           |          |
 register_date | timestamp without time zone |           |          | CURRENT_TIMESTAMP
Indexes:
    "users_pkey" PRIMARY KEY, btree (id)

dfa2lqmmmvka0j=> table users;
 id |       name        |             email             | username | password |       register_date
----+-------------------+-------------------------------+----------+----------+----------------------------
  1 | Francesco Uccelli | francesco.uccelli@utec.edu.pe | Flrotm   | test     | 2019-10-24 21:08:55.128184
  2 | Test1             | f.uccelli.m@gmail.com         | Test1    | test1    | 2019-10-25 08:11:39.573583
  3 | franco            | asdasd                        | asdasd   | asdasd   | 2019-10-25 08:16:22.545654
  4 | FASD              | FASDAD                        | FASD     | FASD     | 2019-10-25 08:23:10.320792
(4 rows)
```



```
                                            Table "public.pedidos"
    Column      |            Type             | Collation | Nullable |                 Default
----------------+-----------------------------+-----------+----------+-------------------------------------
 id             | integer                     |           | not null | nextval('pedidos_id_seq'::regclass)
 nombre_cliente | character varying(36)       |           |          |
 fecha_entrega  | timestamp without time zone |           |          |
 cantidad       | integer                     |           |          |
 cancelada      | character varying(3)        |           |          |
 fecha_pedido   | timestamp without time zone |           |          | CURRENT_TIMESTAMP
 tipo           | integer                     |           |          |
Indexes:
    "pedidos_pkey" PRIMARY KEY, btree (id)

dfa2lqmmmvka0j=> table pedidos;
 id | nombre_cliente |    fecha_entrega    | cantidad | cancelada |        fecha_pedido        | tipo
----+----------------+---------------------+----------+-----------+----------------------------+------
  1 | lanai          | 2019-10-26 00:00:00 |        3 |           | 2019-10-24 22:01:50.390025 |    3
  2 | prueba         | 2019-10-26 00:00:00 |       10 |           | 2019-10-25 05:44:54.162648 |    3
  3 | primero        | 2019-10-26 00:00:00 |       12 |           | 2019-10-25 08:12:20.47883  |    3
  4 | nuevopedido    | 2019-10-26 00:00:00 |       13 |           | 2019-10-25 08:22:50.506645 |    3
(4 rows)
```

Logout



```
@app.route('/logout')
def logout():
    session.clear()
    flash('You are now logged out')
    return redirect(url_for('home'))
```

CONCLUSIONES

- Mejor organización
- Google es tu amigo
- Incremento de dificultad

Referencias

https://wtforms.readthedocs.io/en/stable/crash_course.html#how-forms-get-data

https://www.postgresql.org/docs/9.4/functions-datetime.html

https://docs.sqlalchemy.org/en/13/

bootstraps:

https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css

https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js

https://getbootstrap.com/docs/4.0/components/jumbotron/