```python
# app.py

from flask import Flask, request, jsonify
from models.templates import (
    CurriculumLinkedIn,
    CurriculumEuropass,
    CurriculumMinimalista,
    CurriculumModerno,
)

# Creamos la aplicación Flask
app = Flask(__name__)

# Diccionario para mapear el nombre de plantilla recibido desde el frontend
# con la clase correspondiente en Python
TEMPLATES = {
    "linkedin": CurriculumLinkedIn,
    "europass": CurriculumEuropass,
    "minimalista": CurriculumMinimalista,
    "moderno": CurriculumModerno,
}

@app.route("/preview", methods=["POST"])
def preview():
    """
    Endpoint que recibe los datos del formulario y el tipo de plantilla,
    genera el HTML del CV y lo devuelve al frontend.
    """
    # Obtenemos el JSON enviado desde el frontend
    payload = request.get_json()

    # Extraemos el tipo de plantilla (por defecto, linkedin)
    template_name = payload.get("template", "linkedin")

    # Obtenemos la clase de plantilla correspondiente
    TemplateClass = TEMPLATES.get(template_name, CurriculumLinkedIn)

    # Creamos una instancia de la plantilla
    cv = TemplateClass()

    # Actualizamos los campos del CV con los datos recibidos
    cv.update_from_dict(payload)

    # Generamos el HTML del CV
    html = cv.to_html()

    # Devolvemos el HTML en formato JSON
    return jsonify({"html": html})


@app.route("/ping", methods=["GET"])
def ping():
    """
    Endpoint simple para comprobar que el servidor está vivo.
```

```
    Útil para pruebas rápidas.
    """
    return jsonify({"status": "ok"})


if __name__ == "__main__":
    # Ejecutamos la app en modo debug para desarrollo.
    # En producción, usar gunicorn/uwsgi + Nginx o similar.
    app.run(host="0.0.0.0", port=5000, debug=True)
```