Coding Assignment

Task: implement a symbolic music tokenizer that converts MusicXML files into sequences of discrete tokens suitable for machine learning models.

Practical information:

- Assignment is individual (no groups involved)
- Deadline: February 27th, 23:59 (Hard deadline!)
- Submission through Aula Global
- Format: .ipynb notebook that should run on Google Colab

Coding Assignment

Minimum requirements:

- Score-to-token (tokenizer) script
- No need for token-to-score (de-tokenizer) script (but suggested for testing)

- Tokenizer should:
  - Take as input a MusicXML file
  - Return a list of strings, each of which representing a token
  - Support **any number of parts/instruments**
  - Support **key and time signature changes**
  - Perform tokenization **partwise** (i.e. every parts contains all bars)

Coding Assignment

Tokenizer should handle:

- Beginning/end of sequence tokens: <BOS>, <EOS>
- Parts/Instrumentation: PART_<instrument>
- Clef: CLEF_<type>_<line>
- Pitch: PITCH_<note><octave>
- Position onset (relative to bar): POS_BAR_<onset>
- Position onset (absolute): POS_ABS_<onset>
- Duration: DUR_<quarterLength>
- Rests: REST_<type>
- Bar boundaries: BAR_<measure_number>
- Time signature: TIME_SIG_<num>/<denom>
- Key signature: KEY_<tonic>_<mode>

Coding Assignment

Expected output:

```
[
    "<BOS>",
    # violin part
    "PART_Violin",
    "TIME_SIG_4/4",
    "KEY_C_major",
    "CLEF_G_2",
    # bars of the violín part
    "BAR_1",
    "POS_BAR_0.0", "POS_ABS_0.0", "PITCH_C5", "DUR_1.0",
    "POS_BAR_1.0", "POS_ABS_1.0", "PITCH_D5", "DUR_1.0",
    # ...
]
```