Objetivo:

- Compreender e aplicar a transformação de Sistema de Referência Espacial (SRS) no ambiente de programação Python do QGIS, utilizando a classe `QgsCoordinateTransform`.

Observações importantes:

- Para a realização dos exercícios desta aula, é necessário utilizar o ambiente Python do QGIS, pois diversas classes utilizadas estão disponíveis exclusivamente dentro do QGIS;
- A documentação da biblioteca Python do QGIS está disponível no site oficial: https://qgis.org/pyqgis/3.40. Certifique-se de selecionar a versão correspondente ao QGIS instalado. Nesta aula, adotaremos a versão 3.40;
- No Python, o código é organizado em pacotes (similar a pastas) e módulos (arquivos);
- Todos os módulos do QGIS estão organizados no pacote principal `qgis`, que, por sua vez, contém os seguintes subpacotes principais: `core`, `gui`, `analysis`, `server`, `processing` e `_3d`;
- Cada pacote contém diversas classes com funcionalidades específicas. Nesta aula, utilizaremos classes do pacote `qgis.core`. No Python, utilizamos o ponto (`.`) para indicar hierarquia de pacotes. Exemplo: `qgis.core`.

I. Transformação de Sistema de Referência Espacial (SRS)

Conceito

Transformar o Sistema de Referência Espacial (SRS), também conhecido como **Coordinate Reference System (CRS)**, significa **converter as coordenadas de uma geometria de um sistema para outro**, como, por exemplo, de WGS 84 (EPSG:4326) para SIRGAS 2000 / UTM zona 23S (EPSG:31983).

Essa transformação é realizada por meio da classe **QgsCoordinateTransform**, do pacote qgis.core. A documentação oficial da classe pode ser consultada em: https://qgis.org/pyqgis/3.40/core/QgsCoordinateTransform.html

Construtor utilizado

Embora a classe QgsCoordinateTransform (https://qgis.org/pyqgis/3.40/core/QgsCoordinateTransform.html) possua cinco construtores, utilizaremos o mais simples, que recebe os seguintes parâmetros:

- **Sistema de referência de origem**: objeto do tipo QgsCoordinateReferenceSystem;
- **Sistema de referência de destino**: também um objeto do tipo QgsCoordinateReferenceSystem;
- **Projeto atual**: objeto do tipo QgsProject, que garante a consideração de transformações de datum e outras configurações do projeto QGIS.

Construção dos objetos

Para definir o SRS, utilizamos a classe QgsCoordinateReferenceSystem, que recebe como parâmetro uma **string no formato "EPSG:<código>"**:

```
origem = QgsCoordinateReferenceSystem("EPSG:4326")
destino = QgsCoordinateReferenceSystem("EPSG:31983")
```

- EPSG:4326 - **WGS 84** (sistema de coordenadas geográficas - https://epsg.io/4326);
- EPSG:31983 - **SIRGAS 2000 / UTM zona 23S** (sistema de coordenadas projetadas - https://epsg.io/31983).

Projeto QGIS

Para criar o objeto do projeto atual, utilizamos a classe QgsProject:

```
projeto = QgsProject()
```

⚠ **Importante**: recomenda-se utilizar o método instance() ao invés do construtor direto (QgsProject()), pois este método retorna a instância atual do projeto QGIS em execução. Por exemplo:

```
projeto = QgsProject.instance()
```

Aplicando a transformação

Com os objetos origem, destino e projeto definidos, criamos o objeto de transformação:

```
transformacao = QgsCoordinateTransform(origem, destino, projeto)
```

Para realizar a conversão de coordenadas, utilizamos o método transform() da classe QgsCoordinateTransform. O exemplo a seguir transforma um ponto de Jacareí:

```
origem = QgsCoordinateReferenceSystem("EPSG:4326")
destino = QgsCoordinateReferenceSystem("EPSG:31983")

projeto = QgsProject.instance()

transformacao = QgsCoordinateTransform(origem, destino, projeto)

jcr = QgsPointXY(-45.96642, -23.30555)  # Coordenadas em EPSG:4326
jcr_m = transformacao.transform(jcr)    # Coordenadas transformadas para EPSG:31983

print("WGS 84:", jcr.toString())         # Coordenadas originais
print("SIRGAS 2000:", jcr_m.toString())  # Coordenadas transformadas
```

Resultado:

```
WGS 84: -45.96642, -23.30555
SIRGAS 2000: 401180.885707, 7422325.54167
```

Resumo

Componente	Classe utilizada	Observações
CRS	QgsCoordinateReferenceSystem	Recebe string "EPSG:<código>"
Projeto	QgsProject	Utilize QgsProject.instance()
Transformação	QgsCoordinateTransform	Recebe origem, destino e projeto
Aplicação da transformação	transform()	Aplica a conversão no objeto de coordenadas

Para aprofundamento, recomenda-se a leitura do capítulo **"Projections Support"** do **PyQGIS Developer Cookbook**, disponível em: https://docs.qgis.org/3.40/en/docs/pyqgis_developer_cookbook/crs.html.

Exercícios

Veja os vídeos se tiver dúvida nos exercícios:

Exercícios 1 a 5: https://youtu.be/CYs63WLOXag

Exercícios 6 a 10: https://youtu.be/jzDDubsPReo

Exercício 1: Construa dois objetos com as coordenadas geográficas das cidades de Jacareí e Santa Branca. Em seguida, imprima na tela a distância, em metros, entre elas.

Coordenadas:

- Jacareí: -45.96642, -23.30555
- Santa Branca: -45.88441, -23.39755

Dicas:

- Utilize o construtor `QgsPointXY` para criar um ponto para cada cidade;
- Utilize o construtor `QgsCoordinateReferenceSystem` para definir os sistemas de referência espacial (SRS): de origem (WGS 84) e de destino (SIRGAS 2000);
- Obtenha a instância atual do projeto QGIS em execução `QgsProject.instance()`;
- Crie a transformação de coordenadas com `QgsCoordinateTransform`;
- Use o método `transform()` da classe `QgsCoordinateTransform` para converter as coordenadas dos pontos.
- Use o método `distance()`, da classe `QgsPointXY`, para calcular a distância entre os pontos transformados.

Observação: Certifique-se de transformar ambos os pontos antes de calcular a distância.

Resposta:

```
Distância: 0.12324625795536044
Distância em metros: 13192.307534971656
```

Exercício 2: Construa uma geometria do tipo LineString conectando os pontos das cidades de Jacareí, Santa Branca e Guararema. Imprima na tela a extensão da linha em metros.

Coordenadas:

- Jacareí: -45.96642, -23.30555
- Santa Branca: -45.88441, -23.39755
- Guararema: -46.03542, -23.41555

Dicas:

- Crie os pontos com `QgsPointXY`;
- Transforme os pontos para o SRS desejado antes de criar a linha;
- Use o método `fromPolylineXY()`, da classe `QgsGeometry`, para criar a geometria LineString;
- Utilize o método `length()`, da classe `QgsGeometry`, para obter a extensão da linha.

Observação: A transformação de SRS deve ser feita antes da criação da geometria, pois `transform()` não é aplicável diretamente em `QgsGeometry` para este caso.

Resposta:

```
Extensão em metros: 28751.309092922154
```

Exercício 3: Crie um retângulo utilizando como vértices as coordenadas de Guararema (canto inferior esquerdo) e Jacareí (canto superior direito). Converta o SRS do retângulo de WGS 84 para SIRGAS 2000 (EPSG:31983) e imprima suas coordenadas.

Dicas:

- Utilize o construtor `QgsRectangle(QgsPointXY, QgsPointXY);`
- A conversão de SRS pode ser feita com o método `transformBoundingBox()` da classe `QgsCoordinateTransform`;
- Para obter as coordenadas do retângulo, utilize o método `toString()`.

Resposta:

```
Retângulo: -46.0354200000000020,-23.4155499999999996 : -45.9664199999999994,-23.3055500000000002
Retângulo em metros: 394124.9315238086273894,7410097.8710616137832403 : 401262.3462489063385874,7422325.5416659358888865
```

Exercício 4: Adicione ao código do Exercício 3 as instruções para imprimir a área e o perímetro do retângulo, em metros.

Dicas:

- Utilize os métodos `area()` e `perimeter()` da classe `QgsRectangle`.

Resposta:

```
Área em metros: 85210663.05336808
Perímetro em metros: 38392.69130756636
```

Exercício 5: A partir do código do Exercício 3, adicione instruções para gerar um buffer de 1000 metros ao redor do retângulo e imprimir as coordenadas do retângulo resultante.

Dica:

- Use o método `buffered(distância)`, da classe `QgsRectangle`, com a distância igual a 1000.

Resposta:

```
Retângulo: 394212.2106570758624002,7410097.8710616137832403 :
401180.8857065369375050,7422325.5416659358888865
Alterado: 393212.2106570758624002,7409097.8710616137832403 :
402180.8857065369375050,7423325.5416659358888865
```

Exercício 6: Construa uma geometria do tipo Polygon com os pontos das cidades de Jacareí, Paraibuna, Salesópolis e Guararema. Imprima a área do polígono em metros.

Coordenadas:

- Jacareí: -45.96642, -23.30555
- Paraibuna: -45.673288, -23.373758
- Salesópolis: -45.847091, -23.531665
- Guararema: -46.03542, -23.41555

Dicas:

- Crie os pontos com `QgsPointXY` e transforme-os antes da construção do polígono;
- Utilize `fromPolygonXY()`, da classe `QgsGeometry`, para criar a geometria;
- Calcule a área com `area()` da classe `QgsGeometry`.

Observação: Feche o polígono repetindo o primeiro ponto ao final da lista.

Resposta:

```
Área em metros: 491372625.5691753
```

Exercício 7: A partir do polígono criado no Exercício 6, obtenha o Retângulo Envolvente Mínimo (REM) do polígono e imprima suas coordenadas.

Dicas:

- Utilize o método `boundingBox()` da classe `QgsGeometry`;
- Para exibir as coordenadas, use o método `toString()` da classe `QgsRectangle`.

Resposta:

```
REM: 394212.2106570758624002,7397367.8686728402972221 :
431190.7616127830115147,7422325.5416659358888865
```

Exercício 8: Crie dois retângulos com os pares de coordenadas A-B e C-D. Em seguida, obtenha a interseção entre os retângulos e imprima a área e o perímetro dessa interseção em metros.

Coordenadas:

- A: (0,0)
- B: (2,2)
- C: (1,1)
- D: (3,3)

Dicas:

- Utilize `QgsPointXY` e `QgsRectangle` para criar os retângulos;
- Use `intersect()`, da classe `QgsRectangle`, para obter a interseção;
- Converta o SRS com `transformBoundingBox()` se necessário;
- Use `area()` e `perimeter()` para obter os valores solicitados.

Resposta:

```
Área da interseção em metros: 27575388084.7147
Perímetro da interseção em metros: 664236.4089000672
```

Exercício 9: Crie um polígono com os pontos A, B, C e D. Converta o SRS do polígono para SIRGAS 2000 (EPSG:31983) e imprima a área e o perímetro.

Coordenadas:

- A: (0,0)
- B: (3,0)
- C: (3,3)
- D: (0,3)

Dicas:

- Crie e transforme os pontos antes de construir o polígono com `fromPolygonXY()`;
- Use `area()` e `length()`, da classe `QgsGeometry`, para obter a área e perímetro, respectivamente.

Resposta:

```
Área em metros: 235145229466.8142
Perímetro em metros: 1940052.8679320249
```

Exercício 10: Adicione um anel interno (buraco) ao polígono criado no Exercício 9, utilizando as coordenadas dos pontos E, F, G e H.

Coordenadas:

- E: (1,1)
- F: (2,1)

- G: (2,2)
- H: (1,2)

Dica:

- Utilize o método `fromPolygonXY()` com uma lista de listas: a primeira contendo o anel externo e a segunda contendo o anel interno.

Resposta:

```
Área em metros: 209021014260.91028
Perímetro em metros: 2586589.3780334536
```