

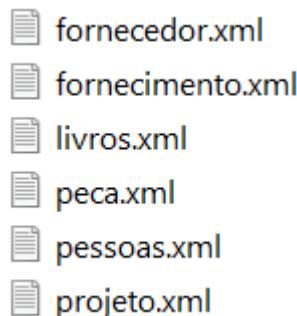
# Trabalho Etapa 1 XML

Exercícios:

Baixe do material de apoio os arquivos XML para esta atividade.

Crie bancos de dados utilizando o software BaseX que também está disponível no material de apoio. É um único arquivo em formato jar.

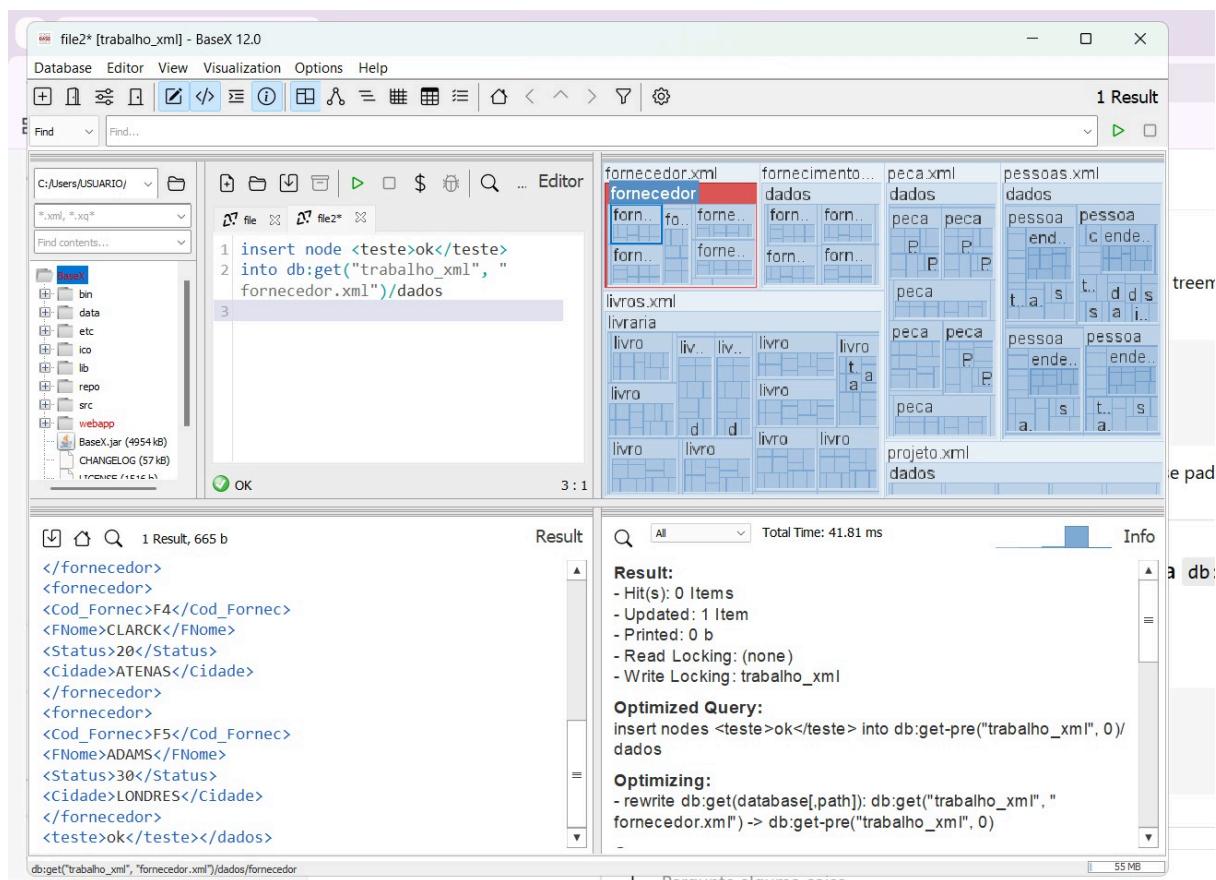
Os arquivos para teste são:



- 1. Etapa 1: A partir dos arquivos anexados, gere instruções Xquery para:**
2. Retornar os dados da penúltima peça da árvore XML.
3. Inserir um atributo com a data em todos os fornecimentos, a data deve ser inserida no formato YYYY-MM-DD.
4. Atualizar o status dos fornecedores de Londres para 50.
5. Retornar o código, a cidade e cor de todas as peças.
6. Obter o somatório das quantidades dos fornecimentos.
7. Obter os nomes dos projetos de Paris.
8. Obter o código dos fornecedores que forneceram peças em maior quantidade.
9. Excluir os projetos da cidade de Atenas.
10. Obter os nomes das peças e seus dados de fornecimento.
11. Obter o preço médio das peças.
- 12. Etapa 2: No PostgreSQL gere o script da base de dados com as tabelas Peca, Fornecedor, Projeto e Fornecimento que está no material de apoio.**

13. Crie uma aplicação em qualquer linguagem de programação que integre os dados da tabela Peca do PostgreSQL com os dados semiestruturados do arquivo Fornecimento.xml, de forma a fazer uma junção entre as duas fontes de dados.
  14. Faça upload dos dois arquivos, das etapas 1 e 2 no repositório do GitHub que está compartilhado com a professora.

Primeiramente eu inseri um nó de teste nos dados dos fornecedores, para ver qual atualização usar:



Depois eu apguei o nó com o comando:

```
delete node db:get("trabalho_xml", "fornecedor.xml")/dados/teste
```

Seguindo para os querys:

1. Retornar os dados da penúltima peça da árvore XML.

The screenshot shows the BaseX 12.0 interface with the following components:

- Top Bar:** Database, Editor, View, Visualization, Options, Help.
- Toolbar:** Includes icons for file operations like Open, Save, Find, and various visualization tools.
- Left Sidebar:** Shows the file system path "C:/Users/USUARIO/" and a list of files including ".xml", ".xq", ".jar", and ".log".
- Central Area:**
  - A code editor window titled "file2\*" containing the XQuery code: `1 db:get("trabalho_xml", "peca.xml")/dados/peca[last() - 1]`.
  - An "Editor" view showing a hierarchical tree of XML documents: fornecedor.xml, fornecimento.xml, peca.xml, and pessoas.xml, each with their respective data nodes.
  - A "Result" view showing the output of the query: 1 Result, 144 b. The result is a single XML node:
 

```
<peca>
<Cod_Peca>P5</Cod_Peca>
<PNome>CAM</PNome>
<Cor>AZUL</Cor>
<Peso>12.00</Peso>
<CIDADE>PARIS</CIDADE>
<Preco>100.00</Preco>
</peca>
```
  - An "Info" panel on the right displaying performance metrics: Total Time: 88.46 ms, Optimized Query, Optimizing, and Compiling details.
- Bottom Status Bar:** Shows "Time required: 88.46 ms" and the user's name "mariomariaquiamaanda".

2. Inserir um atributo com a data em todos os fornecimentos, a data deve ser inserida no formato YYYY-MM-DD.

The screenshot shows the BaseX 12.0 interface with the following components:

- Top Bar:** Database, Editor, View, Visualization, Options, Help.
- Toolbar:** Find, Find... (with dropdown), and a search bar.
- Left Sidebar:** Shows the current file path (C:\Users\USUARIO\file2\* [trabalho\_xml] - BaseX 12.0) and a file browser with a list of files and folders including BaseX.jar, CHangelog, and various XML files.
- Editor Area:** Displays an XQuery script:

```
1 let $data := current-date()
2 for $f in db:get("trabalho_xml",
    "fornecimento.xml")/dados/
    fornecimento
3 return insert node attribute data
    { $data } into $f
```

A red box highlights the third line of the script. A message at the bottom left says "Duplicate attribute name: data." The status bar shows "4 : 1".
- Result Area:** Shows the XML output of the query:

```
<dados>
<fornecimento data="2025-11-18-03:00">
<Cod_Fornec>F1</Cod_Fornec>
<Cod_Peca>P1</Cod_Peca>
<Cod_Proj>J1</Cod_Proj>
<Quantidade>200</Quantidade>
</fornecimento>
<fornecimento data="2025-11-18-03:00">
<Cod_Fornec>F1</Cod_Fornec>
<Cod_Peca>P1</Cod_Peca>
<Cod_Proj>J4</Cod_Proj>
<Quantidade>700</Quantidade>
</fornecimento>
<fornecimento data="2025-11-18-03:00">
```
- Bottom Status Bar:** Shows "76 MB" and a progress bar.
- Bottom Footer:** "Pergunte alguma coisa" (Ask a question).

3. Atualizar o status dos fornecedores de Londres para 50.

file2\* [trabalho\_xml] - BaseX 12.0

Database Editor View Visualization Options Help

Find... Find...

C:/Users/USUARIO/ \*.xml, \*.xq\* Find contents...

Editor

```
for $f in db:get("trabalho_xml", "fornecedor.xml")/dados/
fornecedor[Cidade = "LONDRES"]
return replace value of node $f/
Status with 50
```

OK 3 : 1

fornecedor.xml dados fornecimento... dados peca.xml dados pessoas.xml dados

livros.xml livraria

Result

6 Results, 6829 b

```
<dados>
<fornecedor>
<Cod_Fornec>F1</Cod_Fornec>
<FNome>SMITH</FNome>
<Status>50</Status>
<Cidade>LONDRES</Cidade>
</fornecedor>
<fornecedor>
<Cod_Fornec>F2</Cod_Fornec>
<FNome>JONES</FNome>
<Status>10</Status>
<Cidade>PARIS</Cidade>
</fornecedor>
<fornecedor>
```

Time required: 30.04 ms

mariamariasoulimanda + Pergunte alguma coisa 62 MB

Info

0 Results

5

Total Time: 30.04 ms

- Printed: 0 b
- Read Locking: (none)
- Write Locking: trabalho\_xml

**Optimized Query:**  
db:get-pre("trabalho\_xml", 0)/dados/fornecedor[(Cidade = "LONDRES")] ! replace value of node Status with 50

**Optimizing:**  
- rewrite db:get(database[,path]): db:get("trabalho\_xml", "fornecedor.xml") -> db:get-pre("trabalho\_xml", 0)

**Compiling:**  
- inline for \$f\_0 in db:get("trabalho\_xml", "fornecedor.xml")/dados/fornecedor[(Cidade = "LONDRES")]
- simplify FLWOR expression: db:get("trabalho\_xml", "fornecedor")

4. Retornar o código, a cidade e a cor de todas as peças.

file2\* [trabalho\_xml] - BaseX 12.0

Database Editor View Visualization Options Help

Find Find...

Editor

for \$p in db:get("trabalho\_xml", "peca.xml")/dados/peca  
 return  
 <resultado>  
 { \$p/Cod\_Peca }  
 { \$p/Cidade }  
 { \$p/Cor }  
 </resultado>

**Result**

```
<resultado><Cod_Peca>P1</Cod_Peca><Cidade>LONDRES</Cidade><Cor>VERMELHO</Cor></resultado>
<resultado><Cod_Peca>P2</Cod_Peca><Cidade>PARIS</Cidade><Cor>VERDE</Cor></resultado>
<resultado><Cod_Peca>P3</Cod_Peca><Cidade>ROMA</Cidade><Cor>AZUL</Cor></resultado>
<resultado><Cod_Peca>P4</Cod_Peca><Cidade>LONDRES</Cidade><Cor>VERMELHO</Cor></resultado>
<resultado><Cod_Peca>P5</Cod_Peca><Cidade>PARIS</Cidade><Cor>AZUL</Cor></resultado>
<resultado><Cod_Peca>P6</Cod_Peca><Cidade>LONDRES</Cidade><Cor>VERMELHO</Cor></resultado>
```

Total Time: 4.95 ms

- Printed: 526 b  
 - Read Locking: trabalho\_xml  
 - Write Locking: (none)

**Optimized Query:**  
 db:get-pre("trabalho\_xml", 376)/dados/peca ! <resultado>{ Cod\_Peca, Cidade, Cor }</resultado>

**Optimizing:**  
 - rewrite db:get(database[path]): db:get("trabalho\_xml", "peca.xml") -> db:get-pre("trabalho\_xml", 376)

**Compiling:**  
 - inline for \$p\_0 in db:get("trabalho\_xml", "peca.xml")/dados/peca  
 - simplify FLWOR expression: db:get("trabalho\_xml", "peca.xml")/dados/peca ! <resultado>/ Cod\_Peca Cidade Cor </resultado>

Time required: 4.95 ms

## 5. Obter o somatório das quantidades dos fornecimentos.

File Database Editor View Visualization Options Help

Find... Find...

C:/Users/USUARIO/ BaseX \*.xml, \*.xq\* Find contents...

Editor

```
1 sum(db:get("trabalho_xml", "fornecimento.xml")/dados/fornecimento/Quantidade)
2
3
4
```

OK 2 : 1

fornecedor.xml dados fornecimento.xml dados peca.xml dados pessoas.xml dados

fornecedor.xml dados fornecimento.xml dados peca.xml dados pessoas.xml dados

livros.xml livraria

livro livro livro livro livro livro

livro livro livro livro livro livro

peca peca peca peca peca peca

pessoa pessoa pessoa pessoa pessoa pessoa

projeto.xml dados

1 Result

1 Result, 4 b

1500

Total Time: 5.34 ms

- Printed: 4 b  
- Read Locking: trabalho\_xml  
- Write Locking: (none)

**Optimized Query:**  
sum(db:get-pre("trabalho\_xml", 78)/dados/fornecimento/Quantidade)

**Optimizing:**  
- rewrite db:get(database[,path]): db:get("trabalho\_xml", "fornecimento.xml") -> db:get-pre("trabalho\_xml", 78)

**Query:**  
sum(db:get("trabalho\_xml", "fornecimento.xml")/dados/fornecimento/Quantidade)

Time required: 5.34 ms Pera a um coisa 63 MB

## 6. Obter os nomes dos projetos de Paris.

The screenshot shows the BaseX 12.0 XML editor interface. On the left, there's a file browser with a tree view of files like 'fornecedor.xml', 'fornecimento.xml', 'peca.xml', and 'pessoas.xml'. In the center, a query editor window displays the following XQuery code:

```

1 db:get("trabalho_xml", "projeto.
xml")/dados/projeto[Cidade =
"PARIS"]/Jnome
2
3
4
5

```

The 'Result' pane on the right shows the output of the query, which is a list of names under the 'Jnome' element. Below the results, the 'Info' pane provides performance details and the optimized query plan.

**Result**

1 Result, 21 b

<Jnome>SORTER</Jnome>

Total Time: 1.45 ms

- Printed: 21 b
- Read Locking: trabalho\_xml
- Write Locking: (none)

**Optimized Query:**  
db:get-pre("trabalho\_xml", 685)/dados/projeto[(Cidade = "PARIS")]/Jnome

**Optimizing:**  
- rewrite db:get(database[],path]: db:get("trabalho\_xml", "projeto.xml") -> db:get-pre("trabalho\_xml", 685)

**Query:**  
db:get("trabalho\_xml", "projeto.xml")/dados/projeto[Cidade = "PARIS"]/Jnome

7. Obter o código dos fornecedores que forneceram peças em maior quantidade.

The screenshot shows the BaseX XML database interface. The top navigation bar includes Database, Editor, View, Visualization, Options, and Help. The toolbar contains various icons for file operations like Open, Save, Find, and Print.

The left sidebar displays the file system structure under 'BaseX':

- bin
- data
- etc
- ico
- lib
- repo
- src
- weapp

Below the sidebar are two status bars: 'BaseX.jar (4954 kB)' and 'CHANGELOG (57 kB)'. A 'Time required: 0.28 ms' message is at the bottom left.

The main area consists of several panes:

- Editor:** Shows an XQuery script:

```
1 let $qtds := db:get("trabalho_xml", "fornecimento.xml")/dados/fornecimento/Quantidade
2 let $max := max($qtds)
3 for $f in db:get("trabalho_xml", "fornecimento.xml")/dados/fornecimento[Quantidade = $max]
4 return distinct-values($f/Cod_Fornec)
```
- Result:** Displays the query results in a grid format. The first row is labeled 'livraria.xml' and contains columns for 'livro'. Subsequent rows are labeled 'livraria' and also contain 'livro' columns.
- Performance:** Shows 'Total Time: 9.48 ms' and a small chart.
- Info:** Provides detailed information about the query execution.

Aqui ele forneceu o F1, para garantir que esse era o que fornecia a maior quantidade eu fiz o seguinte:

file2\* [trabalho\_xml] - BaseX 12.0

Database Editor View Visualization Options Help

Find Find...

Editor

```

2 order by xs:integer($f/
Quantidade) descending
3 return
4   <linha>
5     { $f/Cod_Fornec }
6     { $f/Cod_Peca }
7     { $f/cod_Proj }
8     { $f/Quantidade }
9   </linha>
10
11

```

fornecedor.xml  
dados  
for... for... forn... forn... forn... forn...  
fornecimento... dados  
forn... forne... forne... forne... forne...  
peca.xml  
dados  
peca peca P P R R  
pessoas.xml  
dados  
pessoa pessoa end... t... d... s... s... a... i...  
livros.xml  
livraria  
livro livro livro livro t... a...  
livro livro livro livro  
livro livro d... d...  
livro livro

projeto.xml  
dados

Result

```

<linha><Cod_Fornec>F1</Cod_Fornec><Cod_Peca>P1</Cod_Peca
><Cod_Proj>J4</Cod_Proj><Quantidade>700</Quantidade></
linha>
<linha><Cod_Fornec>F2</Cod_Fornec><Cod_Peca>P3</Cod_Peca
><Cod_Proj>J1</Cod_Proj><Quantidade>400</Quantidade></
linha>
<linha><Cod_Fornec>F1</Cod_Fornec><Cod_Peca>P1</Cod_Peca
><Cod_Proj>J1</Cod_Proj><Quantidade>200</Quantidade></
linha>
<linha><Cod_Fornec>F2</Cod_Fornec><Cod_Peca>P3</Cod_Peca
><Cod_Proj>J2</Cod_Proj><Quantidade>200</Quantidade></
linha>

```

Optimized Query:  
(for \$f\_0 in db:get-pre("trabalho\_xml", 78)/dados/fornecimento  
order by xs:integer(\$f\_0/Quantidade) descending empty least  
return <linha>{ \$f\_0/Cod\_Fornec, \$f\_0/Cod\_Peca, \$f\_0/Cod\_Proj,  
\$f\_0/Quantidade }</linha>)

Optimizing:  
- rewrite db:get(database, path): db:get("trabalho\_xml"). "

Pergunte alguma coisa

Aqui conseguimos ver as maiores e menores quantidades

## 8. Excluir os projetos da cidade de Atenas.

The screenshot shows the BaseX 12.0 interface. The top menu bar includes Database, Editor, View, Visualization, Options, and Help. The toolbar contains various icons for file operations like Open, Save, and Print. A search bar with 'Find...' and 'Find...' buttons is present. The main area has tabs for 'Editor' (active), 'file', and 'file2\*'. The 'Editor' tab displays the following XQuery code:

```

1 for $p in db:get("trabalho_xml",
2   "projeto.xml")/dados/projeto[City = "ATENAS"]
3 return delete node $p

```

The results pane shows several XML documents: fornecedor.xml, fornecimento.xml, peca.xml, livros.xml, pessoas.xml, and projeto.xml. The 'projeto.xml' document is highlighted in red at the bottom. Below the results pane, the 'Result' tab displays the XML output of the query:

```

<dados>
<projeto>
<Cod_Proj>J1</Cod_Proj>
<Jnome>SORTER</Jnome>
<Cidade>PARIS</Cidade>
</projeto>
<projeto>
<Cod_Proj>J2</Cod_Proj>
<Jnome>PUNCH</Jnome>
<Cidade>ROMA</Cidade>
</projeto>

<projeto>

```

The 'Info' tab provides performance statistics: Total Time: 12.08 ms, Hits: 0 Items, Updated: 2 Items, Printed: 0 b, Read Locking: (none), Write Locking: trabalho\_xml. The 'Optimized Query' section shows the rewritten query: db:get-pre("trabalho\_xml", 685)/dados/projeto[(City = "ATENAS") ! delete nodes].

## 9. Obter os nomes das peças e seus dados de fornecimento.

The screenshot shows the BaseX 12.0 XML editor interface. The top menu bar includes Database, Editor, View, Visualization, Options, and Help. The toolbar contains various icons for file operations like Open, Save, and Find. A search bar at the top right indicates "4 Results". The main area is divided into several panes:

- Editor pane:** Displays an XQuery script in the "file2\*" database:
 

```

2   $f in db:get("trabalho_xml",
3     "fornecimento.xml")/dados/
4       fornecimento
5     where $p/Cod_Peca = $f/Cod_Peca
6     return
7       <pecaFornecimento>
8         { $p/PNome }
9         { $f/* }
10      </pecaFornecimento>
      
```
- Results pane:** Shows the execution results with 4 results and a total time of 36.72 ms. The results are displayed as XML fragments.
- Database pane:** Shows the structure of the "trabalho\_xml" database, including tables like fornecedor.xml, fornecimento.xml, peca.xml, livros.xml, pessoas.xml, and projeto.xml.
- Log pane:** Displays the optimized query and optimization details.

## 10. Obter o preço médio das peças.

