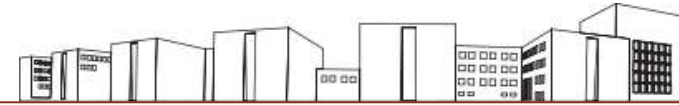


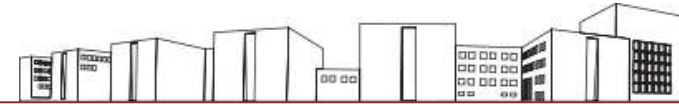
# Notas sobre TinyXML (TP1)

## Laboratório de Aplicações com Interface Gráfica - MIEIC

Rui Rodrigues  
rui.rodrigues@fe.up.pt

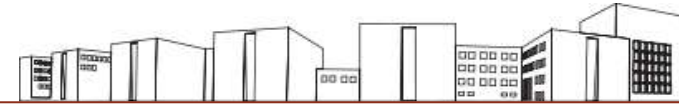


- XML – eXtensible Markup Language
  - Formato textual de dados para codificação de documentos de forma estruturada
  - Usado não só para documentos, mas para representar diferentes estruturas de dados
  - Largamente utilizado nos mais variados contextos, como base para linguagens/estruturas específicas , como RSS, XHTML, SOAP, etc.



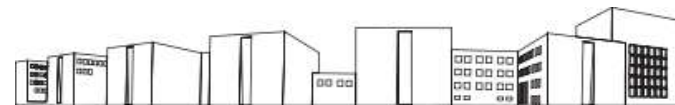
- Tag
  - Uma string rodeada por “<” e “>”
  - Três tipos
    - Start-tag: <HTML>
    - End-tag: </HTML>
    - Empty-element tag <BR/>
- Element
  - Um componente do documento
    - <H1>Título</H1>
    - <HR />
  - Pode ter outros elementos aninhados (filhos/children)
    - <DIV>
      - <B>Nome</B>
    - </DIV>
- Attributes
  - Pares nome=“valor”, definidos numa start-tag ou empty-element-tag
    - <a href=“**www.fe.up.pt**” target=“**\_blank**”>FEUP</a>
    - <br id=“**break**”/>

# XML como árvore

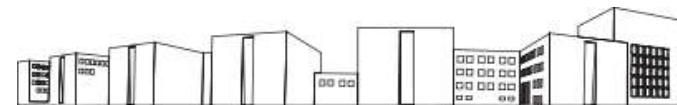


- Um documento XML, dada a possibilidade de aninhamento, pode ser representado como uma árvore
- Como tal, também pode ser percorrido como uma árvore, navegando de pai para filho e vice-versa, e entre irmãos (siblings)

# Mais sobre XML

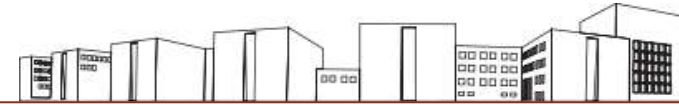


- Boa-formação
- Schemas e validação – DTD/XSD, etc.
- Conversão – XSLT
- Pesquisa - XPath

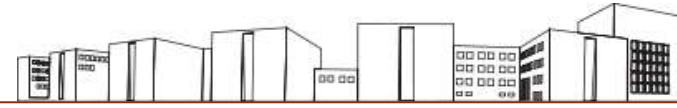


- Uma biblioteca C++ que implementa um parser de XML e permite (entre outras funcionalidades):
  - a leitura de um documento XML para uma estrutura interna de dados constituída por nós e elementos (Nodes e Elements) XML,
  - o acesso aos dados de diversas formas: iterativamente ou através de acesso directo a determinados elementos
  - a alteração da estrutura, inserindo, modificando ou removendo elementos
  - a gravação para ficheiro em formato XML de uma estrutura em memória

## Classes importantes



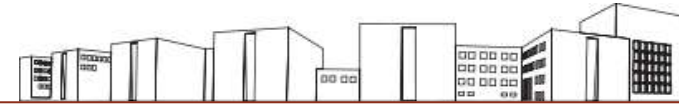
- TiXmlNode – Um nó da árvore do documento
  - FirstChildElement([tag name])
    - O primeiro elemento descendente deste nó cujo tipo de tag é “tag name”
  - NextSiblingElement([tag name])
    - O próximo irmão deste nó cujo tipo de tag é “tag name”
  - Value()
    - O nome da tag deste nó



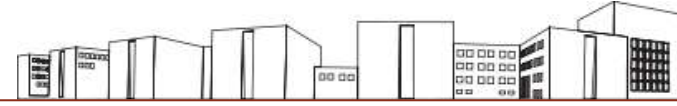
- TiXmlDocument (subclasse de TiXmlNode)
  - Constructor(filename)
    - Inicializa a estrutura interna do documento com o nome do ficheiro associado
  - LoadFile()
    - Carrega o ficheiro associado pelo construtor
    - O objecto fica como nó de raíz do documento
    - Devem ser validados os possíveis erros

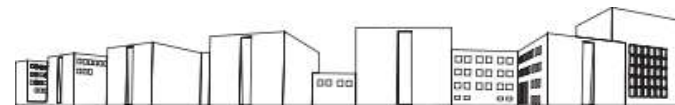


## Classes importantes



- TiXmlElement (subclasse de TiXmlNode)
  - Attribute(name)
    - Retorna a string correspondente ao valor associado ao atributo “name”
    - Erros de atributo inexistente devem ser validados
  - QueryIntAttribute(name, int \* value)
    - Métodos para simplificar a leitura de valores
    - Erros devem ser validados (==TIXML\_SUCCESS)
    - Existem variantes para outros tipos (e.g. Float)





- Começar no nó de raiz, identificar os nós de cada secção, processar cada secção na sua função/classe
- Podem usar map's para armazenar inicialmente os objectos/texturas/materiais, mas devem guardar também apontadores para não acederem aos maps em tempo de rendering
- Pré-calculas as matrizes de transformação de cada nó, e guardar apenas a matriz
- Aproveitar a recursividade de funções para implementar a stack de materiais e texturas
- Usar a linha de comando para indicar o ficheiro XML como parâmetro