

## Tipos de Elementos de Implementação

Os Elementos de Implementação podem geralmente ser categorizados como arquivos e diretórios. Os arquivos podem ser ainda mais categorizados da seguinte forma:

<b>Arquivos de código-fonte</b>	<p>Existem arquivos que podem ser modificados diretamente por um desenvolvedor. Isso inclui:</p> <ul style="list-style-type: none"><li>• código fonte compilável (como: arquivos .h, .cpp e .hpp para C++, CORBA IDL ou .java para Java)</li><li>• origem interpretada (HTML, várias linguagens de script) e arquivos de dados modificáveis pelo usuário (como tabelas de banco de dados, arquivos de configuração, arquivos gráficos e outros).</li></ul>
<b>Arquivos derivados</b>	<p>Estes arquivos <b>não</b> destinam-se a ser modificados diretamente por um desenvolvedor. Eles são derivados dos arquivos de origem.</p> <p>Alguns arquivos derivados destinam-se ao uso no sistema operacional (como DLLs, JARs e EXEs). Outros são os resultados intermediários da compilação (por exemplo, arquivos .o da compilação de C++ e arquivos .class da compilação de código java).</p>

## Modelando Elementos de Implementação

Na versão 1.3 da especificação UML (Unified Modeling Language), os arquivos são modelados como componentes da UML. No UML 1.4, a representação foi alterado para ser artefato UML. Muitas ferramentas e perfis UML continuam utilizando os componentes para modelar arquivos.

Em qualquer caso, a modelagem de arquivos e diretórios deve ser feita moderadamente, a menos que exista algum suporte automatizado. Os arquivos podem ser visualizados na estrutura de diretórios do projeto e o relacionamento entre os arquivos e os elementos de design é geralmente suficientemente claro na estrutura de diretórios e nas convenções de nomenclatura.