

# Trabalho 3 - AVE

Ambientes Virtuais de Execução

1718SI – LI41N - Grupo 8

39081 – Paulo Piriquito ; 42186 – João Francisco

## Introdução

Nesta fase do trabalho adicionámos à classe `HtmlEmit` o suporte para métodos de tipo genérico para podermos parametrizar através de delegates para qualquer tipo do modelo um formato do html desejado, evitando desta forma recorrer à reflexão que é realizada quando é utilizada uma instancia simples de `HtmlEmit`.

## Metodologia

Conforme pedido no enunciado do trabalho adicionámos os métodos:

- `ForTypeDetails<T>>(Func<T, string> transf)`
- `ForTypeInTable<T>(IEnumerable<string> headers, Func<T, string> transf)`
- `ForSequenceOf<T>(Func<IEnumerable<T>, string> transf)`

Estes métodos retornam uma nova instância de `HtmlEmit` em que os seus métodos `ToHtml(object source)` e `ToHtml<T>(IEnumerable<T> arr)` retornam a string html resultante da execução do delegate que foi passado na instanciação do `HtmlEmit`.

Este comportamento é obtido à custa `ForTypeDetails`, `ForTypeInTable` e `ForSequenceOf` guardarem num dicionário `HtmlFormatterTypes<Type, IHttpGetter>` à parte do utilizado na implementação simples um novo objeto do tipo `HtmlFormatObj<T> : IHttpGetter`, (que por sua vez serve de contentor para a função delegate do parâmetro), e que quando chamado o método `ToHtml` é verificado para evitar a inclusão de html que seria nesta fase adicionado ao retorno do `ObjPropertiesToHtml`.

O `ObjPropertiesToHtml` também tenta obter para um dado tipo um `IHttpGetter` deste novo dicionário para tipos customizados, caso não haja nenhuma instancia presente correspondente a esse tipo executa o algoritmo já presente da fase anterior que seria o de

tentar obter a partir do dicionário `htmlTypes`, em que no caso de também não existir nenhuma instancia armazenada para o tipo pretendido é emitida uma nova através de emissão de código IL.

## Testes

Para testar as novas funcionalidades foi criada a classe de testes `HtmlEmitFormattedTest` que compreende os testes `HtmlFormatterDetailsTest`, `HtmlFormatterForTypeInTableTest` e `HtmlFormatterForSequenceOfTest` em que cada um é executado o `htmlEmit.For...` com a respetiva expressão para a função do delegate e é feito um assert comparando o resultado da obtenção do html com o html esperado definido á partida da classe de testes.

## Conclusão

Conclui-se então que a utilização de tipos genéricos associados a delegates permitem um maior nível de abstração no código, assim como maior flexibilidade na adaptação de uma classe para suportar várias formas de parametrização e extensão das funcionalidades, assim como uma forma mais eficiente de filtrar elementos de estruturas de dados e trata-los de forma concreta para cada tipo diferente.

Desta forma é possível também em casos em que seriam utilizados processos de reflexão, evitar fazê-lo ao suportar um tipo genérico que é tratado da forma que foi parametrizada pela classe que faz a utilização.