



# AVE

2015/16 Verão

## Série Exercícios 2

---

### Grupo 1

Henrique Calhó nº38245

Edgar Demétrio nº39393

João Pereira nº37628

### Docente

Miguel Gamboa Carvalho

## Objectivo

Tal como na primeira série é pretendido que se serialize objetos json, mas para reduzir o uso de reflexão é usada instrumentação de código.

## Implementação

A nossa implementação consiste na utilização de dois dicionários onde guardamos a informação dos tipos para um dado critério que já foram emitidos.

Um dicionário de pares String, ICriteria, que verifica se o objecto recebido já foi emitido para ser serializado para a ICriteria passado como parâmetro. Se foi é chamado o método da interface retornada após a emissão para serializar o objeto, se não emitimos o código responsável por serializar o objecto de acordo com o critério passado.

Um outro dicionário é criado em EmmitterUtils de pares Type, IEmmitterCriteria de modo a saber que código emitir dado o critério selecionado.

Na projeto da solução o package Reflect é referente à primeira série de exercícios:

**Jsonfier.cs** - responsável pela serialização de objetos

**Icriteria.cs** - interface para implementações de critérios de serialização

**FilterByField.cs**, **FielByProperty.cs** e **FilterParametlessNonNoidMethod.cs** são as implementações feitas por nós.

A única alteração feita nestas classes foi a adição do método ToJsonDictionary(object) para podermos reutilizar o código da série 1 para facilitar a serialização de tipos primitivos e string. Caso estejamos a lidar com um tipo complexo é redirecionado para a verificação de emissão do código para esse objeto.

No package Instr estão as classes referentes à instrumentalização de código:

**Jsoninstr.cs** - classe responsável pela instrumentação de um dado tipo com um dado critério. Verifica se este já existe no dicionário e executa a implementação da interface

**EmmitterUtils** - classe utilitaria para gerar um tipo que implementa a interface ICriteria cujo metodo execute ao qual vão ser adicionadas instruções IL

**IEmmitterCriteria** - interface para implementações de critérios de instrumentação

**EmitterProcessFields**, **EmmitProcessProperty** e **EmitterProcessParametlessNonVoidMethod** são as implementações suportadas.

## Testes

Para testar o funcionamento da emissão de código para os critérios implementados adicionámos à classe de testes **TestJsonfier.cs** os métodos

**TestEmmitVsReflectionWithPropertyForStudent**,  
**TestEmmitVsReflectionWithFieldsForStudent** e  
**TestEmmitVsReflectionWithParametlessNonVoidReturnForStudent**.

Para verificar que a chamada de métodos pela emissão de código é mais rápida que a chamada por reflexão foi criada uma amostra de quinze chamadas em que verificámos o tempo de cada uma. Validando assim as conclusões propostas pelo enunciado.