

Article Title

JOSÉ MORGADO
pg27759@alunos.uminho.pt

LUÍS MIGUEL PINTO
pg27756@alunos.uminho.pt

PEDRO CARNEIRO
pg25324@alunos.uminho.pt

Resumo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

I. INTRODUÇÃO

Tem-se assistido a um crescimento da complexidade das arquiteturas dos sistemas de software, sendo cada vez mais comum a integração de múltiplos componentes. Hoje em dia, ninguém desenvolve aplicações web sem recorrer a frameworks e web services.

De modo a que isto seja possível, precisamos de técnicas que permitam garantir a modularidade e extensibilidade dos componentes de software. Foi neste contexto que surgiram os conceitos de Inversão de Controlo e Injeção de Dependências.

II. INVERSÃO DE CONTROLO

A Inversão de Controlo é um princípio de concepção de software. Tipicamente, quando queremos construir um programa apenas codificamos parte do mesmo e recorremos a bibliotecas para realizar parte das tarefas; neste caso, o nosso código (específico) depende de código externo (reutilizável). Usando Inversão de Controlo, a abordagem é a oposta: o código reutilizável das bibliotecas é que vai necessitar/depende do nosso código específico.

Um exemplo disso mesmo são as frameworks. Estas coordenam a execução das

aplicações, o que nos permite ter que definir apenas parte do comportamento das mesmas. Ou seja, funcionam como esqueletos aos quais apenas temos de disponibilizar os plugins necessários.

Há várias técnicas que seguem este princípio, como a Injeção de Dependências, que será apresentada na secção seguinte.

III. INJEÇÃO DE DEPENDÊNCIAS

Maecenas sed ultricies felis. Sed imperdiet dictum arcu a egestas.

- Donec dolor arcu, rutrum id molestie in, viverra sed diam
- Curabitur feugiat
- turpis sed auctor facilisis
- arcu eros accumsan lorem, at posuere mi diam sit amet tortor
- Fusce fermentum, mi sit amet euismod rutrum
- sem lorem molestie diam, iaculis aliquet sapien tortor non nisi
- Pellentesque bibendum pretium aliquet

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facili-

sis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

REFERÊNCIAS

[Figueredo and Wolf, 2009] Figueredo, A. J. and Wolf, P. S. A. (2009). Assortative pairing and life history strategy - a cross-cultural study. *Human Nature*, 20:317–330.