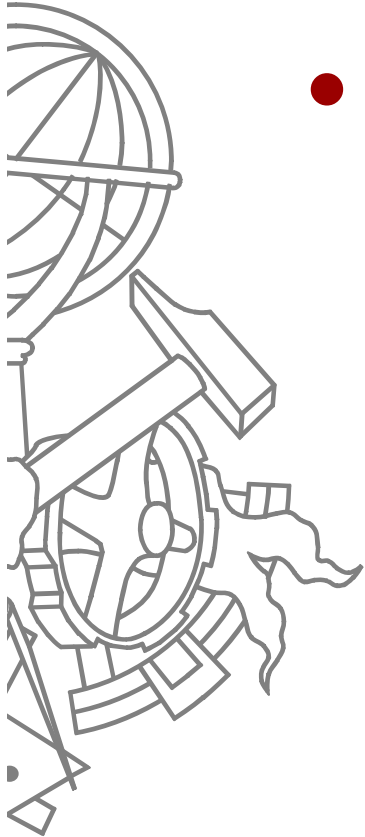


Arquitectura de Aplicação & modelo 3 camadas

Paulo Sousa

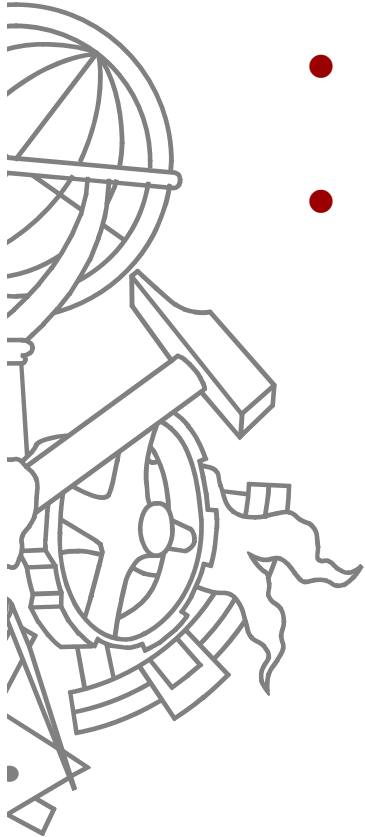
Engenharia da Informação
Instituto Superior de Engenharia do Porto

O que é Arquitectura de Aplicação?



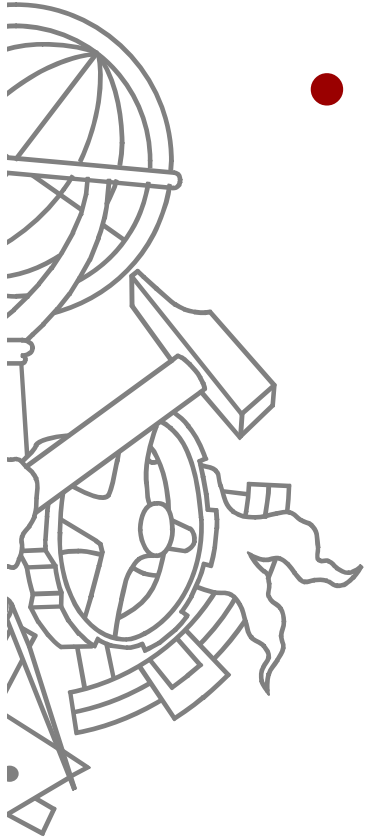
- A set of... design elements that have a particular form.
 - Perry, D.E. & Wolf, A.L. "Foundations for the Study of Software Architecture." Software Engineering Notes, ACM SIGSOFT 17, 4 (October 1992): 40-52.

O que é Arquitectura de Aplicação?



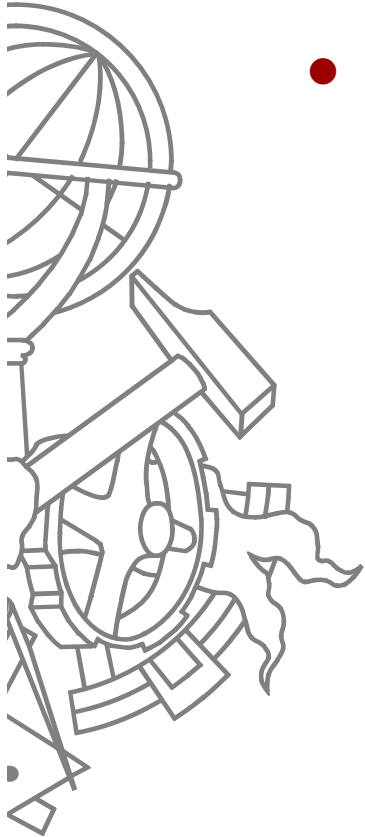
- [a level of design concerned with **issues**] **beyond the algorithms and data structures** of the computation;
- designing and specifying the overall system structure emerges as a new kind of problem. Structural issues include gross organization and global control structure; protocols for communication, synchronization, and data access; assignment of functionality to design elements; physical distribution; composition of design elements; scaling and performance; and **selection among design alternatives**.
 - Garlan, D. & Shaw, M. "An Introduction to Software Architecture," in Advances in Software Engineering and Knowledge Engineering, vol. I. River Edge, NJ: World Scientific Publishing Company, 1993.

O que é Arquitectura de Aplicação?



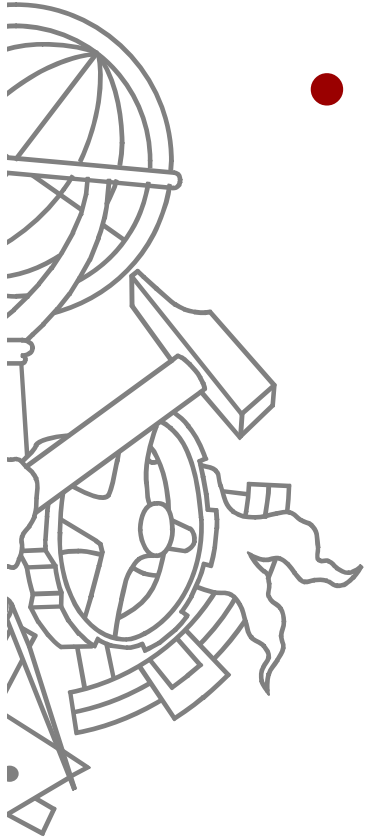
- The **structure** of the components of a program/system, their interrelationships, and **principles and guidelines** governing their design and evolution over time.
 - Garlan, David & Perry, Dewayne. *Introduction to the Special Issue on Software Architecture*. IEEE Transactions on Software Engineering 21, 4 (April 1995).

O que é Arquitectura de Aplicação?



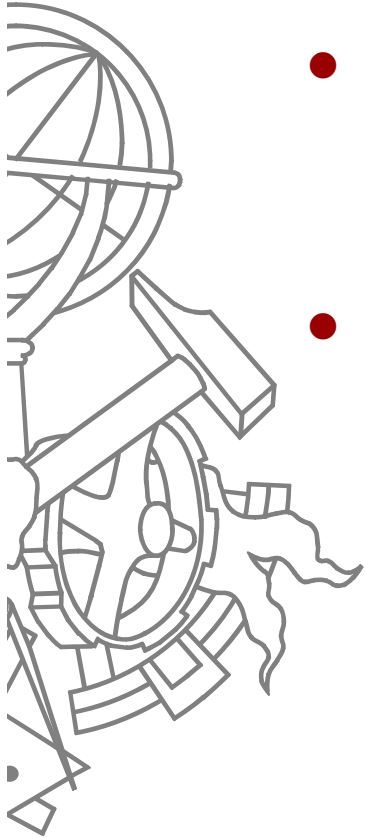
- The set of **significant decisions** about the **organization** of a software system, the selection of the **structural elements and their interfaces** by which the system is composed, together with their **behavior** as specified in the collaborations among those elements, the composition of these structural and behavioral elements into progressively larger subsystems, and the architectural **style that guides this organization** - these elements and their interfaces, their collaborations, and their composition
 - Booch, Rumbaugh, and Jacobson, The UML Modeling Language User Guide, Addison-Wesley (1999).

O que é Arquitectura de Aplicação?



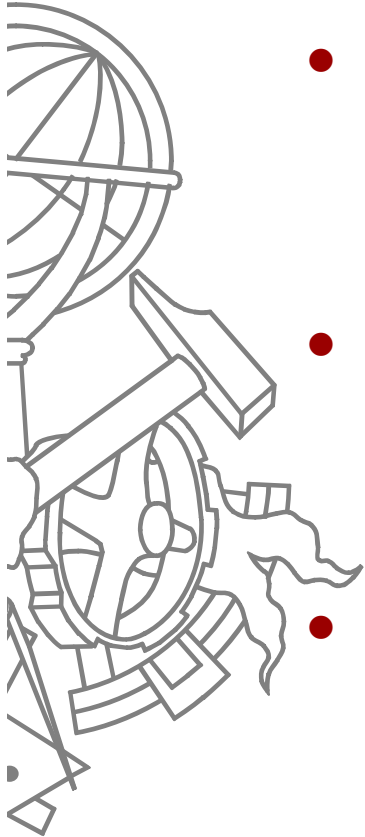
- The fundamental **organization** of a system, embodied in its components, their relationships to each other and the environment, and the **principles governing its design and evolution.**
 - ANSI/IEEE Std 1471-2000, Recommended Practice for Architectural Description of Software-Intensive Systems

O que é Arquitectura de Aplicação?



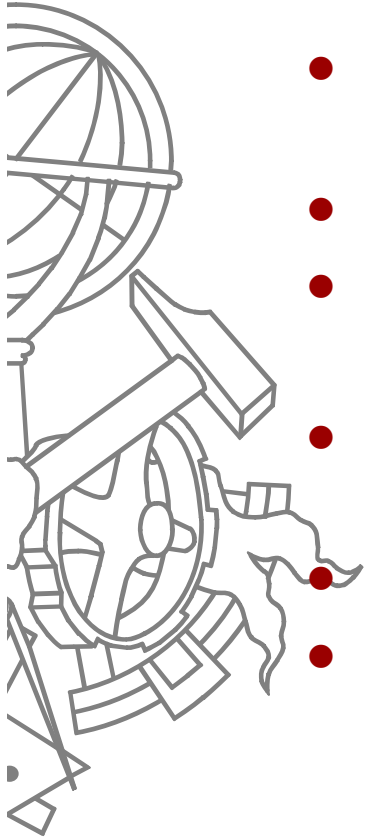
- The architecture of any automated services that **support and implement functional requirements**, including the interfaces to the business and other applications.
- It describes the **structure** of an application and how that structure implements the functional requirements of the organization. Whilst there should ideally be one application architecture in an organization, **there are typically many different app. architectures.**
 - Microsoft Architecture Overview, <http://msdn.microsoft.com/architecture/overview/default.aspx?pull=/library/en-us/dnea/html/eaarchover.asp>

O que é Arquitectura de Aplicação?



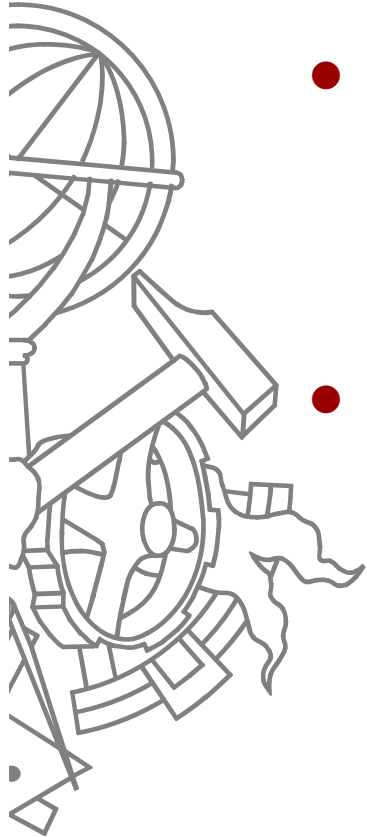
- The software architecture of a program or computing system is the **structure** or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.
- By "externally visible" properties, we are referring to those assumptions other components can make of a component, such as its **provided services**, performance characteristics, fault handling, shared resource usage, and so on.
- The intent of this definition is that a software architecture **must abstract away some information from the system** (otherwise there is no point looking at the architecture, we are simply viewing the entire system) and yet provide enough information to be a basis for analysis, decision making, and hence risk reduction
 - Software Architecture in Practice (2nd edition), Bass, Clements, Kazman; Addison-Wesley (2003)

Ou seja...



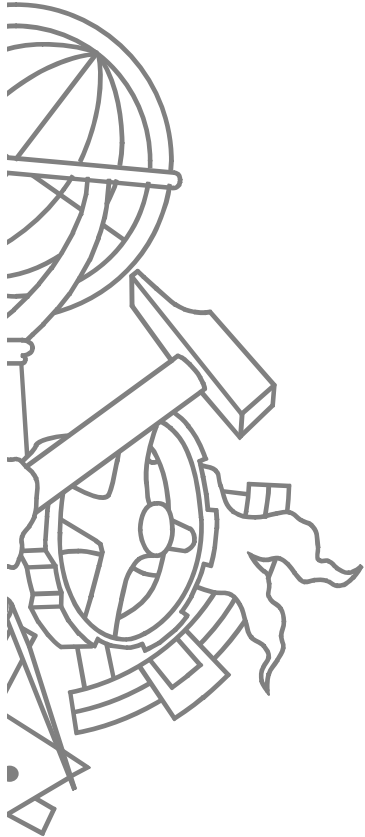
- Organização de alto nível e estrutura (relações) de componentes do sistema
- Princípios (regras) de desenho
- Decisões fundamentais
 - Difíceis de alterar posteriormente
- Fundação para o desenvolvimento e evolução do sistema
- Suporta e implementa funções de negócio
- Deve permitir a evolução do sistema
 - Estrutura
 - Requisitos
 - Funcionalidades
- Existem várias arquitecturas (ou vistas) no sistema

Para que serve?



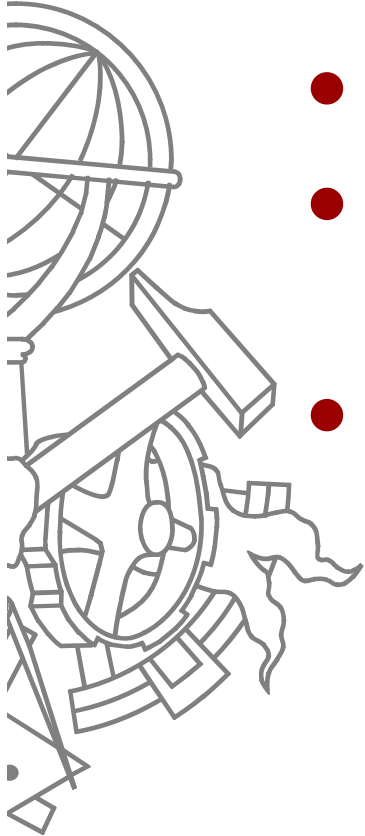
- **Perspectiva organizacional**
 - Comunicação do desenho de alto nível
 - Fornece contexto ao sistema
 - Alocação de trabalho
- **Perspectiva técnica**
 - Cumprir requisitos e objectivos
 - Potencia flexibilidade
 - Potencia redução de custos de manutenção e evolução
 - Aumenta a reutilização e integração com sistemas *legacy* e componentes de terceiros

Arquitectura de aplicação



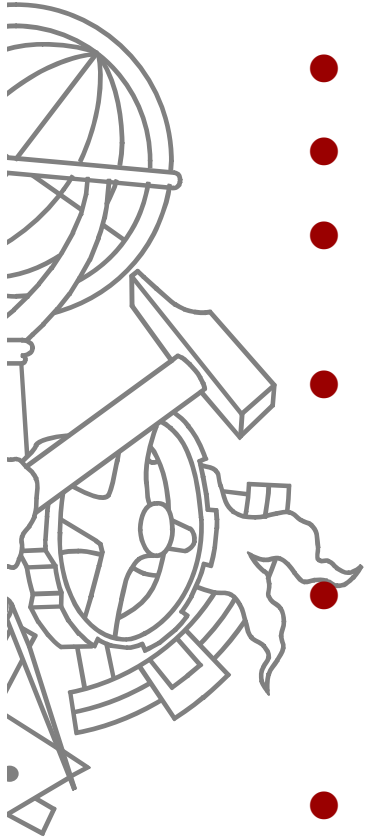
É difícil dizer se uma arquitectura de aplicação é correcta ou errada. Uma arquitectura resulta de um conjunto de compromissos e concessões entre várias forças e do contexto específico.

O que não é uma arquitectura de aplicação



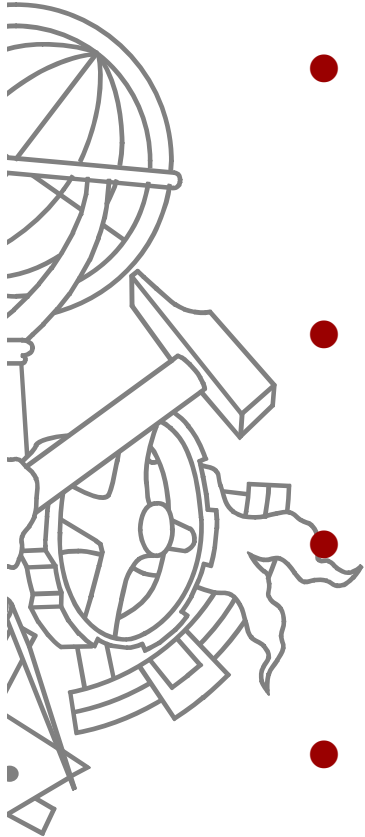
- Modelo de dados
- Arquitectura física onde o sistema vai executar
- Detalhes de desenho/implementação

Rules of thumb



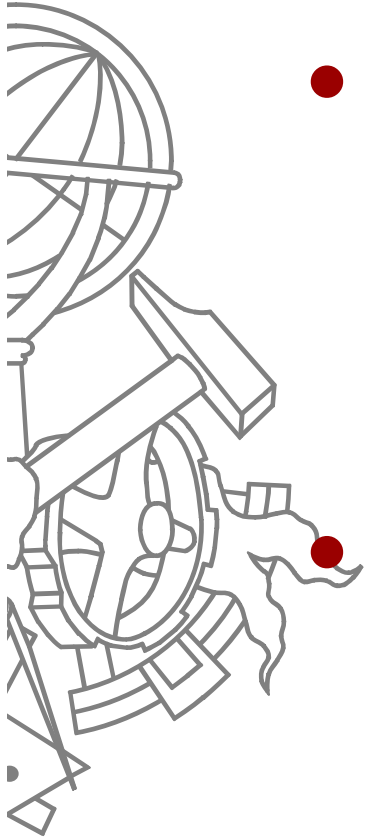
- Elementos isolados (componentes)
- Responsabilidades bem definidas
- Comunicam através de interfaces bem definidas
- Diminuir as dependências entre elementos
 - Programar para a interface e não para a implementação
- Tentar antecipar a mudança/evolução
 - Por exemplo, através de parametrização e carregamento dinâmico de componentes
- Ligações assíncronas sempre que possível
 - Por exemplo, *Message Queueing*

Exemplos



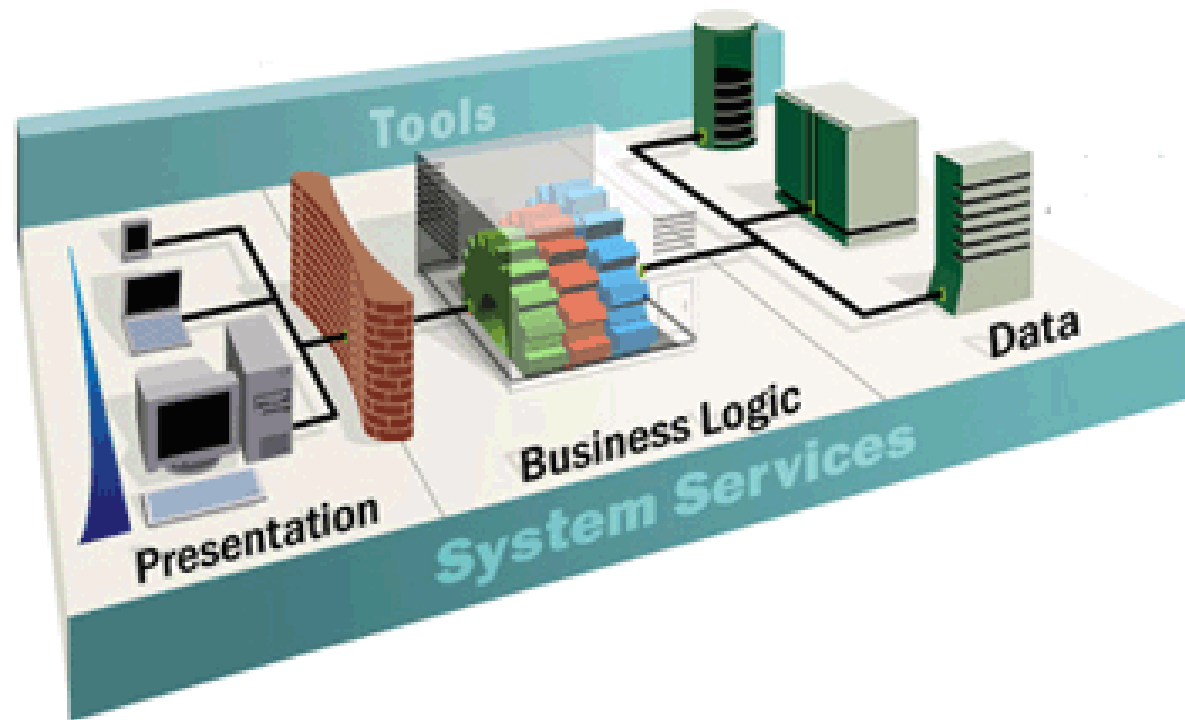
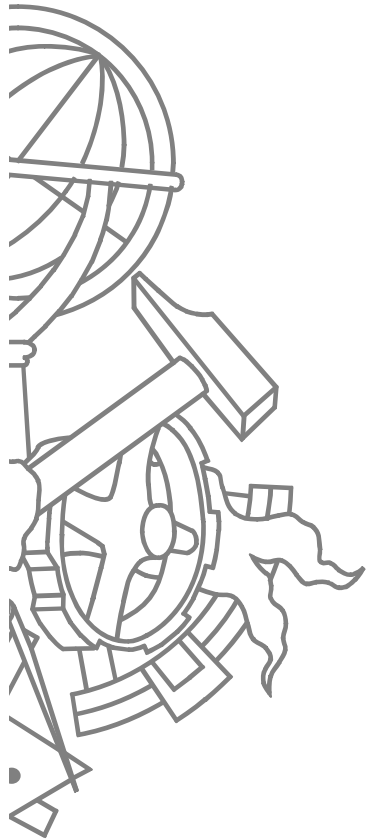
- PetStore (J2EE)
 - http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/sample-app/sample-app1.3.1a3.html
- PetShop (.Net)
 - <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/petshop3x.asp>
- FoodMovers (.Net)
 - <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvsent/html/FoodMovers0.asp>
- AdventureBuilder (J2EE)
 - <http://java.sun.com/blueprints/code/adventure/1.0/docs/architecture.html>

Um caso concreto

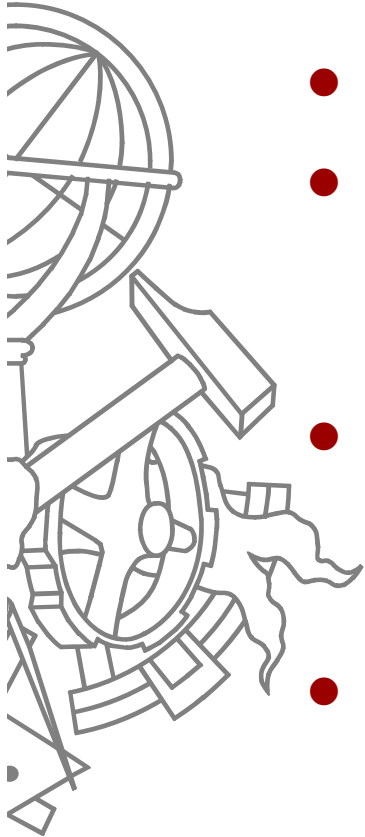


- Um exemplo de arquitectura de aplicação (mais correctamente, de meta-arquitectura ou padrão) é o modelo 3 camadas
- “Meta-arquitectura” porque não dá uma solução em concreto para um caso específico mas sim um modelo, um arquétipo, que pode ser aplicado a vários casos

Arquitetura de três camadas (3-layer)

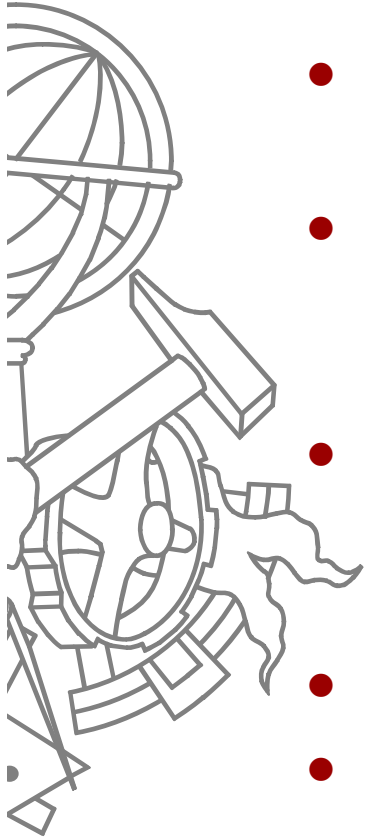


3-layer



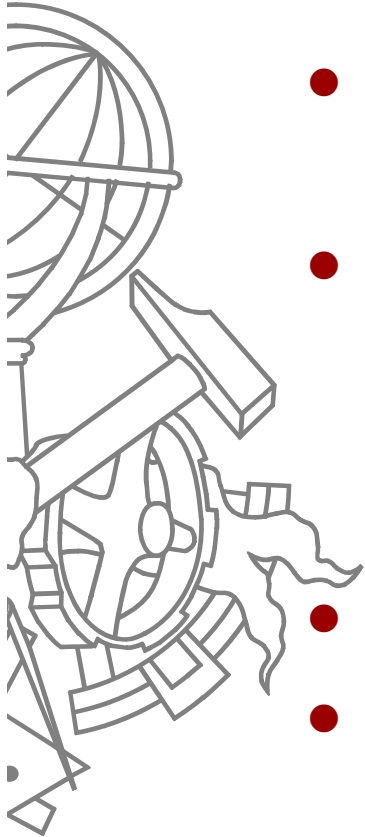
- O sistema é dividido em camadas separadas
- Serviços de Apresentação
 - Fornecem a interacção entre o utilizador (ou sistema externo) e a aplicação
- Regras de Negócio
 - Representam o núcleo da aplicação em termos de processamento
- Serviços de Dados
 - Fornecem serviços de persistência

3-layer



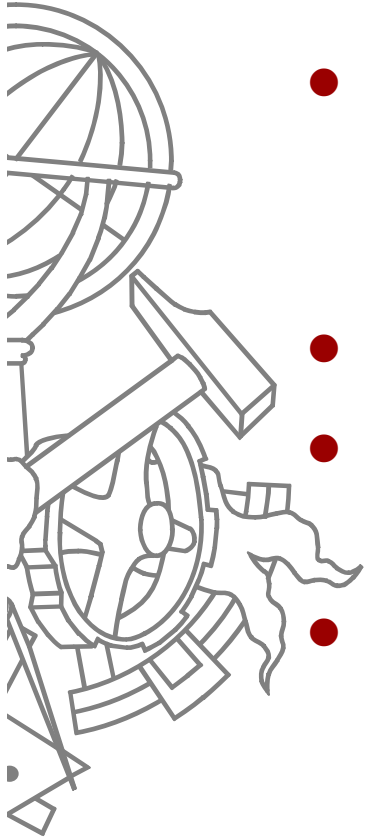
- Normalmente cada camada está num componente isolado
- Cada camada apenas depende da camada seguinte
 - Normalmente não são admitidos “saltos” por cima de uma camada (ex., da 1ª para a 3ª)
- Permite alterações em qualquer uma das camadas sem interferência nas outras
 - Desde que se mantenham as interfaces das classes
- Potencia separação de responsabilidades
- Permite especialização da equipa de desenvolvimento
- Potencia a facilidade de manutenção do código
- Pode aumentar a complexidade de compreensão do código

A 1ª camada: Apresentação



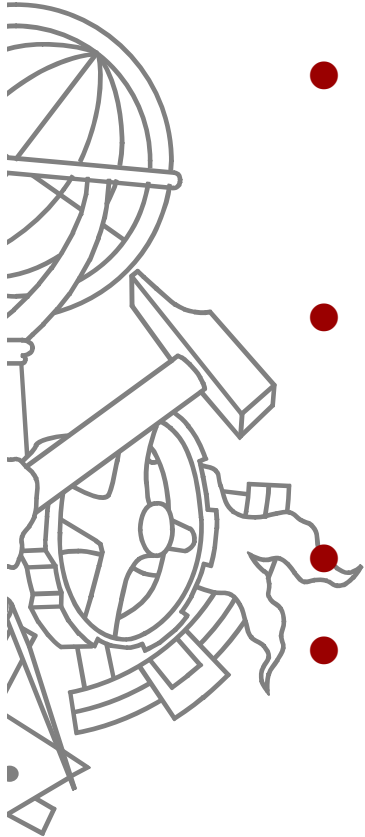
- Apenas se preocupa com a interacção com o utilizador
- Possui lógica de apresentação
 - Tratamento de inputs
 - Visualização de resultados
 - Validações que aumentem a usabilidade
- Pode ser um programa não interactivo (*batch*)
- Permite implementar filosofia **multi-canal**
 - Exemplo: uma livraria online pode adicionalmente fornecer serviços através da TVinteractiva modificando apenas a camada de apresentação e reutilizando todos os componentes de negócio

A 2ª camada: Lógica de negócio



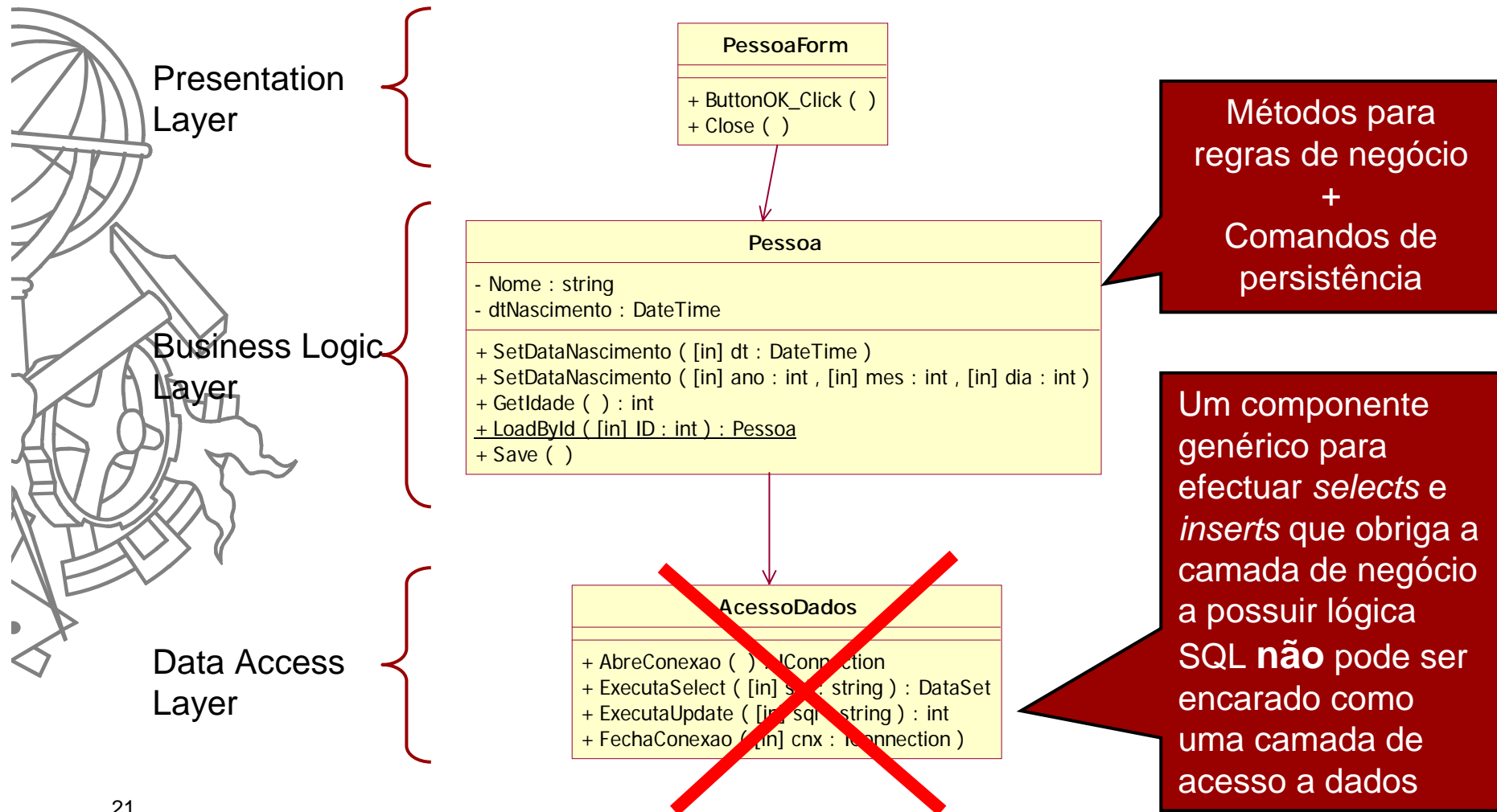
- Apenas se preocupa com a implementação das regras de negócio associadas ao problema e às entidades de negócio
- Desconhece “quem” é a aplicação cliente
- Implementa todas as validações de dados que necessitar
- Desconhece os pormenores de persistência dos dados
 - Recorre aos serviços da camada de acesso a dados

A 3ª camada: Serviços de dados

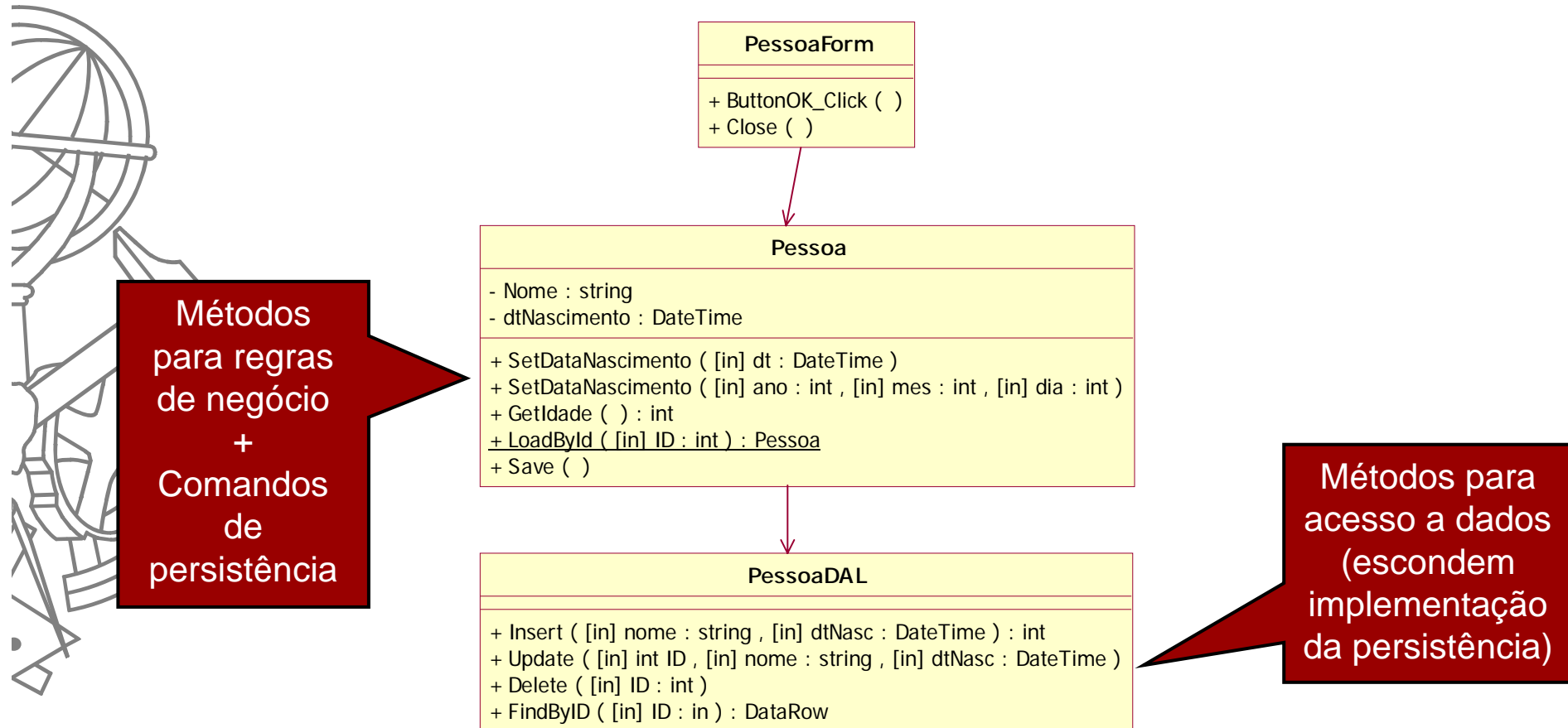


- A 3ª camada **não** são os dados, nem os servidores de base de dados, mas sim os **serviços de acesso** aos dados
- Encapsulam, de um ponto de vista de operações de negócio, o acesso aos dados, isolando a camada de lógica de negócio
- Strings de SQL **só** devem existir nesta camada
- Tecnologias tipo ADO.net ou JDBC **não** são a 3ª camada
 - Poderão ser consideradas uma 4ª camada ou subcamada

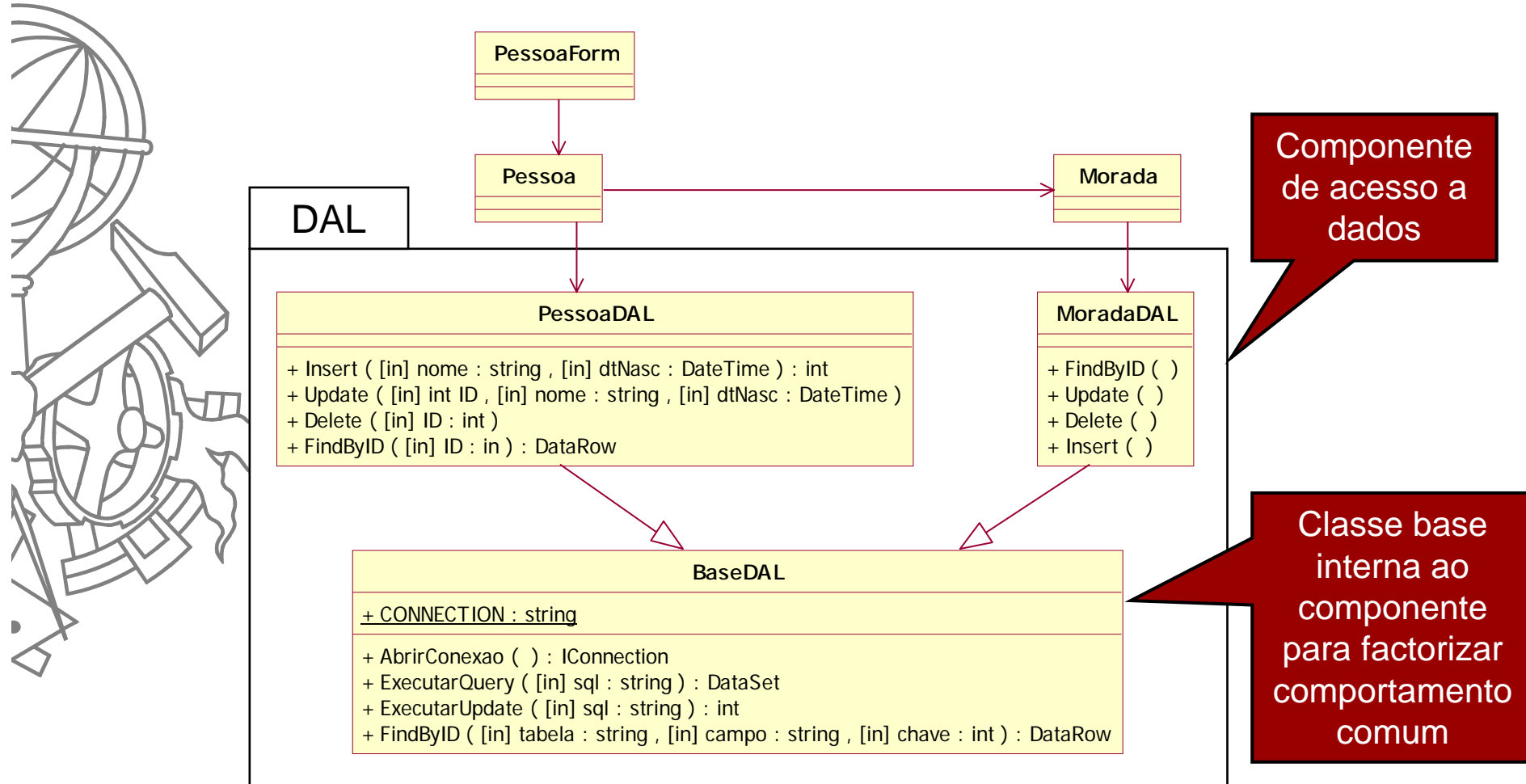
Exemplo de 3 camadas



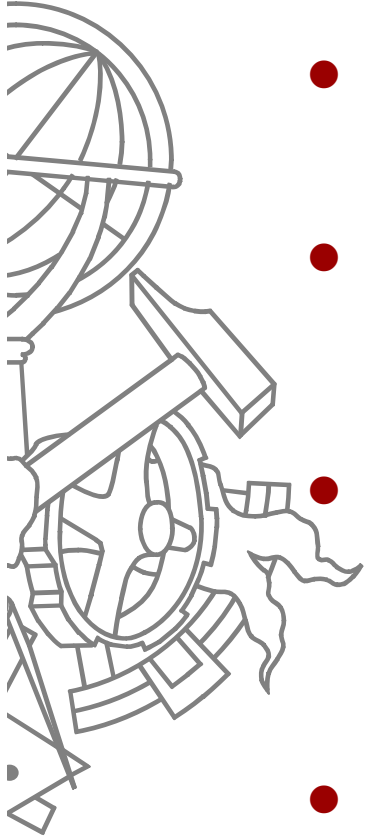
Exemplo de 3 camadas



Exemplo de 3 camadas

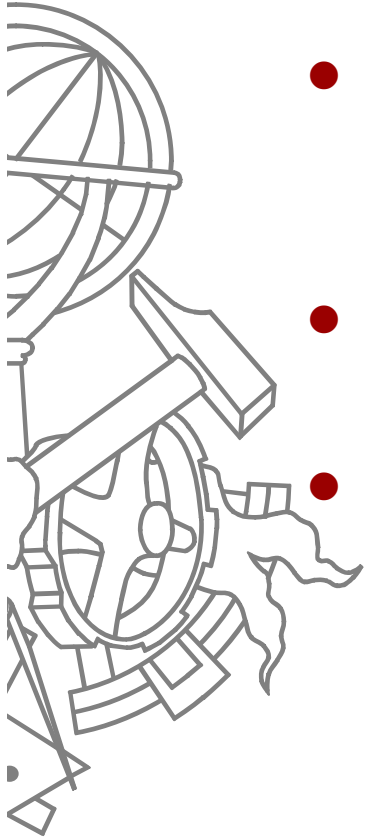


Layers vs. Tiers



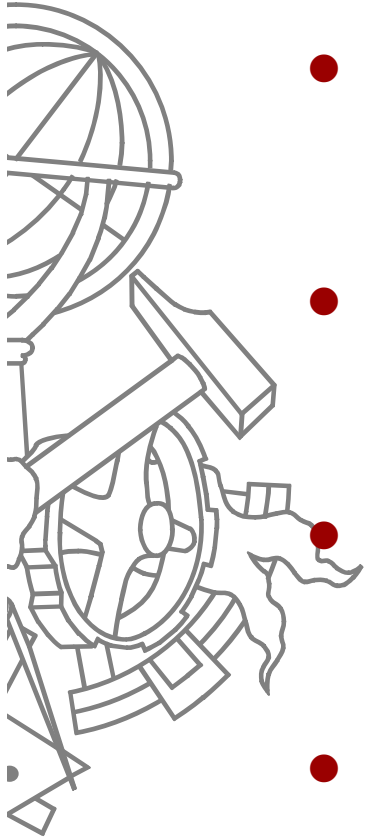
- Alguma ambiguidade na literatura...
- Layer
 - Camada lógica de funcionalidade
 - Layers referem-se à organização do código e dos dados
- Tier
 - Camada física de instalação (*deployment*)
 - Tiers referem-se à distribuição do código e dos dados
- n-layer não implica n-tier
- n-tier (infelizmente) não implica n-layer

N-tiers



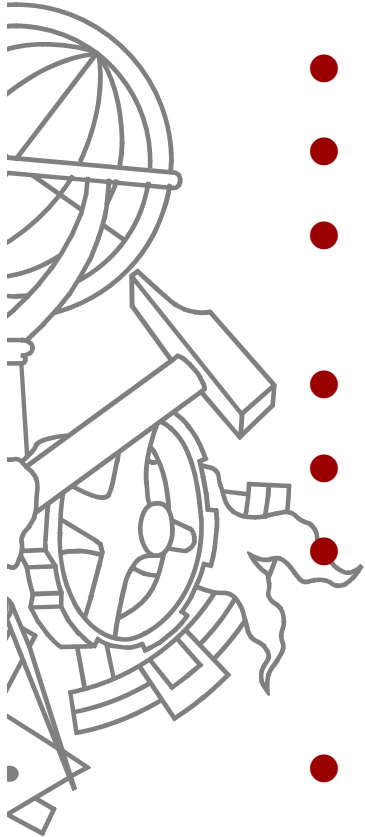
- Permite uma grande escalabilidade da aplicação se as classes não mantiverem estado
 - *Load balancing*
- Permite a distribuição de carga da aplicação por diferentes máquinas
- Permite a evolução de sistemas legados ao inclui-los numa das camadas
 - provavelmente com a utilização de um *Proxy* para interligação
- *Vamos voltar a este assunto mais tarde...*

Em que contexto?



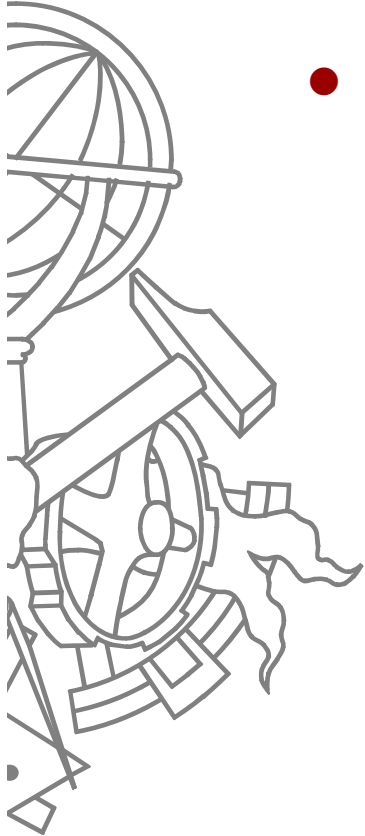
- Divisão por camadas dá mais trabalho a desenvolver
- Não vale a pena o trabalho para “somar dois números”
- Mas vale a pena para qualquer aplicação mais complexa
- Principalmente para **aplicações empresariais**
 - O que são?

Aplicação empresarial

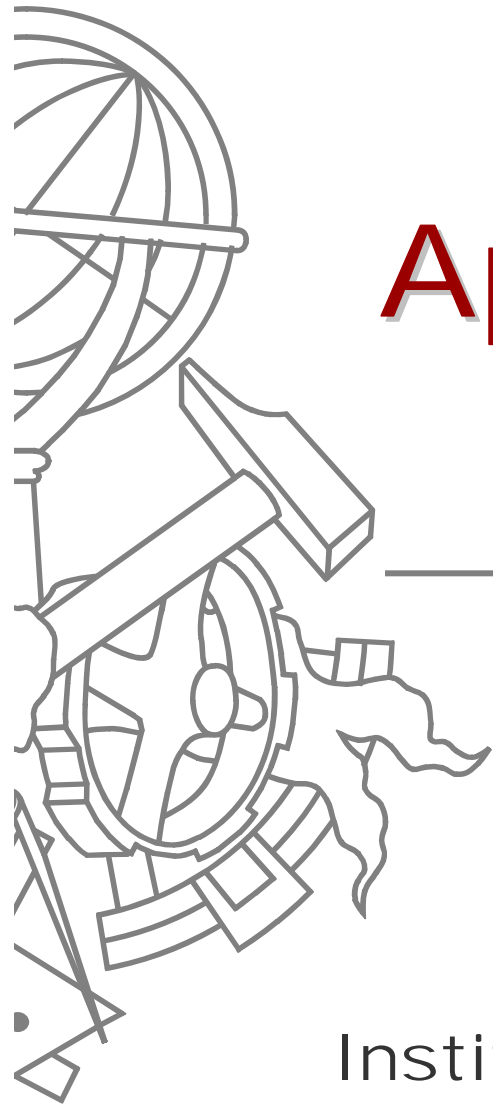


- Funcionalidade crítica para a empresa
- Dados persistentes e em grande quantidade
- Acessos simultâneos e concorrentes a esses dados
- Grande número de ecrãs
- Integração com outras aplicações
- Dissonância conceptual entre conceitos com mesmo nome entre departamentos/unidades (aplicações) diferentes
- Regras de negócio complexas e por vezes “ilógicas” (para contemplar situações particulares)

E então?



- Como é que podemos no nosso desenvolvimento tirar partido da experiência e das boas práticas identificadas por outros?
- **Padrões de software**



Arquitectura de Aplicação & modelo 3 camadas

Paulo Sousa

Engenharia da Informação
Instituto Superior de Engenharia do Porto