

# Sistemas Distribuídos

Rui Raposo

Thursday 21<sup>st</sup> May, 2020  
16:47

## 1 Modelos de Programação Distribuída

1. Modelos de comunicação por mensagens (**comunicação por *Sockets***);
2. **Modelos Arquiteturais.**

A arquitetura de um sistema distribuído é a estrutura do sistema em termos de **localização** das suas diferentes partes, do **papel** que cada parte desempenha e como se interrelacionam.

A arquitetura tem implicações no **desempenho, fiabilidade e segurança do sistema.**

### 1.1 Modelos Arquiteturais

Camadas de um sistema distribuído:

- Aplicações e serviços;
- *Middleware*, camada de *software* que tem como objetivo mascarar a heterogeneidade de um sistema distribuído fornecendo um modelo de programação uniforme;
- Sistema Operativo (**Plataforma**);
- Computador e hardware da rede (**Plataforma**).

#### 1.1.1 Modelo – Cliente Servidor

Modelo independente do *middleware* utilizado, **modelo assimétrico.**

**Servidor** : processo passivo que quando contactado por um cliente envia uma resposta.

**Cliente** : contacta o servidor com o objetivo de utilizar um serviço; envia um pedido e fica à espera da resposta.

**Cliente e servidor são papéis que podem ser desempenhados.** Uma entidade pode simultaneamente ser cliente e servidor. Um processo para responder a um pedido, pode ter que recorrer a outro serviço, sendo cliente deste.

Num sistema de informação típico, existem **três** classes de funcionalidades:

- Camada de apresentação – Responsável pela *interface* com o utilizador;
- Camada de lógica de negócios – Controlam o **comportamento** da aplicação;
- Camada de persistência de dados – Assegura o **armazenamento e integridade dos dados**.

Servidor em **3 camadas**:

- *Web browser* – HTML;
- Servidor *Web* – PHP, Java;
- Base de Dados – MySQL, SQLServer, oracle;

### 1.1.2 Cliente/Servidor replicado

Existem vários servidores, capazes de responder aos mesmos pedidos.

**Vantagens:**

- Permite distribuir a carga, melhorando o desempenho;
- Não existe um ponto de falha único.

**Problema:**

- Manter o estado do servidor coerente em todas as réplicas.

### 1.1.3 Cliente/Servidor particionado

Existem vários servidores com a mesma *interface*, cada um capaz de responder a uma parte dos pedidos, **DNS**.

Servidor redirige o cliente para outro servidor (**iterativo**).

Servidor invoca o pedido noutro servidor (**recursivo**).