41951 Análise de Sistemas - 2024/25 [ASis25]

**Paulo Lacerda (120202), João Leite (119859), Ellen Sales (117450), Rui Albuquerque (110509)**

Grupo 2, vApr 7, 2025

Relatório do Lab 09

[Introdução](#_nf2dfutrlpma)

[Distribuição do trabalho](#_3g0uzqz2jjf6)

[Referências e materiais consultados](#_x3k8jx13h2cd)

[Atividades](#_d7avh5axs3a)

[Atividade 1](#_r392zx2gb3mg)

[Atividade 2](#_g3cgcdeqw33)

[Atividade 3](#_5vakp1a2tfyh)

[Atividade 4](#_blycnwgwcqc2)

# Introdução

## Distribuição do trabalho

A coordenação deste lab, no nosso grupo, foi assegurada por Ellen Sales.

O trabalho resultou das seguintes contribuições:

* Atividade 1 - Ellen Sales
* Atividade 2 - Paulo Lacerda
* Atividade 3 - João Leite
* Atividade 4 - Rui Albuquerque

## Referências e materiais consultados

# Atividades

## Atividade 1

| Categoria | Requisito com impacto | Porque é importante? |
| --- | --- | --- |
| Extensibilidade | Integração com restaurantes deve ser baseada em interfaces bem definidas. | Interfaces bem definidas permitem a integração de serviços externos sem comprometer os módulos existentes permitindo, por exemplo, incluir novos restaurantes ou funcionalidades de *analytics*. |
| Escalabilidade | O sistema deve suportar um milhão de utilizadores em simultâneo | Para garantir um desempenho estável e fluido mesmo durante picos de utilização (almoço e jantar), a arquitetura precisa ser altamente escalável, utilizando por exemplo *load balancers, auto-scaling e microserviços* distribuídos. |
| Desempenho/Tempo de resposta | O sistema deve garantir respostas atempadas e um tempo de indisponibilidade mínimo, mesmo durante os períodos de maior tráfego. | A lentidão pode levar à frustração dos utilizadores e perda de vendas. Uma arquitetura orientada a eventos e *caching* eficiente pode mitigar atrasos, especialmente no carregamento de menus e finalização de encomendas. |
| Usabilidade e consistência entre plataformas | A aplicação deve apresentar uma experiência consistente entre diferentes plataformas (web, app cliente, app entregador, dashboard restaurante) | A consistência melhora a experiência do utilizador e reduz erros de utilização. Uma abordagem com *componentes reutilizáveis* e *design system* ajuda a manter a uniformidade entre interfaces |
| Extensibilidade | Integração com restaurantes deve ser baseada em interfaces bem definidas. | Permite a integração de diferentes sistemas (tablets, smartphones, POS), facilitando a adição de novos restaurantes sem reestruturar o sistema central |
| Segurança e conformidade | Informações pessoais e dados de pagamento devem ser cifrados e o sistema deve estar em conformidade com o RGPD | A violação de dados pode causar danos à reputação e penalizações legais. A arquitetura deve incluir mecanismos de *data encryption*, *tokenização*, *segurança em APIs* e gestão de consentimento. |
| Disnponibilidade/Recuperação de falhas | Cada componente deve ser capaz de recuperar rapidamente de falhas, mantendo a alta disponibilidade | Interrupções afetam diretamente as vendas e a confiança do utilizador. Arquiteturas resilientes com *circuit breakers*, *fallbacks* e *monitorização* ativa são fundamentais. |

## 

## Atividade 2

| **Requisito com Impacto na Arquitetura** | **Medida Técnica / Recomendação** | **Justificação / Nexo de Causalidade** |
| --- | --- | --- |
| Integração com restaurantes deve assentar em interfaces bem definidas | **1. API REST Padronizada**  - Endpoints claros (ex: /menus, /pedidos)  - Verbos HTTP (GET, POST, PUT, DELETE)  - Formato JSON para os dados transmitidos | Facilita a interoperabilidade entre sistemas heterogéneos, seguindo boas práticas universais de integração. |
|  | **2. Documentação com OpenAPI/Swagger**  - Especificação detalhada dos endpoints, estruturas de dados e exemplos | Reduz ambiguidades e acelera o desenvolvimento por parte dos restaurantes parceiros. |
|  | **3. Versionamento da API**  - Utilização de URLs versionadas (ex: /v1/pedidos) | Garante compatibilidade retroativa durante a evolução do sistema. |
|  | **4. Autenticação e Segurança**  - HTTPS obrigatório  - Autenticação via OAuth 2.0 ou API Keys | Protege dados sensíveis e assegura que apenas parceiros autorizados têm acesso. |
|  | **5. Gateway de API**  - Utilização de ferramentas como AWS API Gateway ou Kong para gestão centralizada | Facilita o encaminhamento, monitorização e aplicação de políticas (ex: controlo de taxa de requisições). |
|  | **6. Webhooks para Eventos**  - Notificações em tempo real (ex: actualizações de pedidos) | Elimina a necessidade de sondagens constantes (polling), reduzindo carga desnecessária no sistema. |
|  | **7. Validação de Estruturas de Dados**  - Utilização de JSON Schema para validar os dados recebidos | Garante a consistência dos dados trocados entre os sistemas. |
|  | **8. Monitorização e Registo (Logging)**  - Ferramentas como Prometheus + Grafana para métricas em tempo real | Permite detecção rápida de falhas e gargalos, aumentando a fiabilidade da plataforma. |

## 

## 

## 

## 

## 

## 

## 

## 

## Atividade 3

* Módulo de Autenticação:
  + Login/Registo;
  + Sessão de convidado;
* Módulo de Catálogo de Restaurantes e Menus:
  + Listagem de restaurantes e filtros;
  + Detalhes dos menus;
  + Personalização de pratos;
* Módulo de Carrinho e Checkout:
  + Adição/remover itens;
  + Aplicação de cupões;
  + Integração com pagamentos;
* Módulo de Pagamentos:
  + Integração com gateways externos;
  + Segurança (cifração, RGPD);
* Módulo de Encomendas:
  + Gestão de encomendas;
  + Estado da encomenda (preparação, entrega, etc.);
  + Localizador de encomendas em tempo real;
* Módulo do Restaurante:
  + Dashboard do restaurante;
  + Receção e preparação de encomendas;
  + Estimativa de tempo;
  + Suporte para tablets/POS;
* Módulo de Entregador:
  + App do entregador parceiro;
  + Gestão de tarefas e rotas;
  + Atualização de estado;
* Módulo de Recomendação:
  + Sugestões personalizadas com base em histórico;
* Módulo de Relatórios e Analytics:
  + Desempenho de menus;
  + Tendências de vendas;
* Módulo de Infraestrutura e Escalabilidade:
  + Alta disponibilidade;
  + Balanceamento de carga;
  + Recuperação de falhas;

## 

## 

## 

## Atividade 4

### Tipos de Arquitetura

No livro de Richards, são discutidos três estilos de arquitetura, nomeadamente:

1. Big Ball of Mud: Refere-se a sistemas sem uma estrutura arquitetural clara, desenvolvidos de forma desordenada e com soluções improvisadas. Este tipo de arquitetura resulta em código difícil de modificar, testar e escalar.
2. Client/Server (Dois Níveis): Trata-se de uma separação entre o cliente (interface de utilizador) e o servidor (processamento e armazenamento de dados). Esta arquitetura é utilizada em sistemas onde a comunicação entre os dois componentes é essencial, como em aplicações de comércio eletrónico ou sistemas de gestão de dados.
3. Arquitetura de Três Níveis: Composta por três camadas distintas: o banco de dados, o servidor de aplicação e o cliente (interface de utilizador). Este estilo permite uma maior organização e escalabilidade do sistema, tornando-o adequado para aplicações mais complexas.

### Relacionamento com os Trabalhos A e B

* Trabalho A (Integração de Comércio Eletrónico e Back-Office): Este sistema pode ser associado à arquitetura Client/Server, pois a loja online funciona como cliente, enquanto o back-office, responsável pela gestão de produtos e processamento de encomendas, opera como servidor. A separação clara entre o cliente e o servidor permite uma arquitetura eficaz para este tipo de sistema, especialmente com a implementação em nuvem usando serviços como Cloud Functions e Firestore.
* Trabalho B (Automação de Cozinhas): Este sistema se relaciona mais com a arquitetura Três Níveis, uma vez que envolve camadas distintas para gerenciar o inventário de ingredientes, realizar previsões de consumo e interagir com o utilizador (administradores, manipuladores de alimentos, gestores de loja). O uso de IA e IoT adiciona complexidade ao sistema, permitindo uma separação clara entre a base de dados, os processos de aplicação e a interface com o utilizador, o que é típico da arquitetura de três níveis.