41951 Análise de Sistemas - 2024/25 [ASis25]

Lab 09 - Documentação da arquitetura

Feb 10, 2025 | Atividades na definição e documentação de uma arquitetura.

## Percurso de aprendizagem

Em grupo.

### Decisões de arquitetura

A determinação de requisitos estabelece o âmbito do que é para fazer, mas não inclui decisões sobre a melhor forma de construir o sistema.

Essa parte fundamental na conceção da solução técnica que inclui a tomada das decisões que melhor equilibram o custo de criar uma solução e a manter, e as qualidades necessárias do sistema, corresponde ao trabalho da arquitetura do software.

As decisões de arquitetura são baseadas nos requisitos, especialmente nos requisitos “de qualidade” com impacto na definição de arquitetura.

* Estude o caso apresentado na secção “[Caso de estudo](#_1p0z7rfw4tai)” neste documento, relativo à implementação de uma plataforma de encomenda de comida online.
* Considere, principalmente, os requisitos não funcionais que se depreende do texto. Apresenta uma lista de requisitos com impacto na arquitetura (5+) conforme o exemplo:

| 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*. |
|  |  |  |
|  |  |  |

Tópicos como os seguintes elementos poderão ajudar na análise:

* A intensidade de utilização é estável?
* Existe mais que uma *fontend?* Até que ponto devem ser consistentes entre si?
* A aplicação requer a utilização abundante de multimédia, como as imagens dos pratos?
* Como poderá a aplicação do estafeta ajudar à sua produtividade?
* O seguimento em tempo real do pedido é necessário?
* Os pedidos em tempo-real podem colocar o sistema num estado inconsistente?
* Existem regulamentos (segurança, privacidade,...) a considerar?
* Os utilizadores toleram demoras ocasionais na utilização das aplicações ou isso é uma fonte de frustração que pode fazê-los desistir?
* etc.

### Recomendações para uma arquitetura compatível

Retome os requisitos que identificou antes.

Discuta, com ajuda dos colegas, de recursos online, ou do docente, que opções técnicas de desenho[[1]](#footnote-0) (arquitetura) podiam ser consideradas para satisfazer o requisito.

Note que há um grande grau de liberdade nos resultados que pode apresentar, mas deve ser possível estabelecer um nexo de causalidade entre entre recomendação para o desenho técnico e a capacidade pretendida.

Depois, cada grupo deverá apresentar as suas conclusões à turma.

| Requisito com impacto na arquitetura | Medida técnica/Recomendação |
| --- | --- |
| Integração com restaurantes deve ser baseada em interfaces bem definidas. | Utilização de API REST expondo serviços para a integração com os Sistemas de Informação dos restaurantes (menus, pedidos). |
|  |  |
|  |  |

### Vista de arquitetura

A arquitetura é, de modo geral, a partição criteriosa de um todo em módulos que comunica a organização e um sistema e permite estruturar o trabalho cooperativo entre grupos de pessoas para resolver um complexo

Um “módulo”, neste contexto, encapsula um conjunto coeso de funcionalidade, e não implica nenhuma tecnologia de implementação ou distribuição. O objetivo é comunicar quais são os “blocos” principais da solução e interação entre eles (comunicação).

* A partir do caso de estudo, elabore um diagrama que procure mostrar os módulos significativos da solução.

Nota: pode usar a [notação da UML](#_jhd3bdy3j28y), mas não é estritamente necessário.

### Estilos de arquitetura

* Veja os três primeiros estilos de arquitetura discutidos no livro de Richards ([capítulos 10, 11, 12 e 17](https://learning.oreilly.com/library/view/fundamentals-of-software/9781492043447/part02.html)). Não precisa de estudar em profundidade; o objetivo é ficar com uma ideia geral.
* Consulte, no repositório de dissertações da UA, o [trabalho A](http://hdl.handle.net/10773/36293) (relacionado com a integração do comércio eletrónico e processos de back-office de uma empresa) e o [trabalho B](http://hdl.handle.net/10773/44659) (automação de cozinhas). Estude a descrição da arquitetura presente.
* Com qual dos estilos de arquitetura é que a organização sa soliuação dos trabalhos A e B se podem relacionar, repetivamente? Justifique

## Caso de estudo

A *Bifanas.come* é uma plataforma para encomendar refeições em restaurantes locais, para recolha (*pick-up*) ou entrega (*home delivery*). Incluir sempre o requisito “o sistema deve suportar um milhão de utilizadores em simultâneo”.

Os utilizadores podem navegar por diversos menus, personalizar pratos e fazer pagamentos seguros, tudo numa aplicação ou site conveniente. O sistema deve gerir o elevado tráfego durante as horas de ponta dos restaurantes, como os períodos de ponta do almoço e do jantar, garantindo respostas atempadas e um tempo de indisponibilidade mínimo.

Assim que os clientes iniciam sessão (ou continuam como convidados), podem explorar uma variedade de cozinhas, ver itens do menu detalhados e adicionar itens a um carrinho virtual. As preferências dietéticas especiais (por exemplo, sem glúten, vegan) estão disponíveis como filtros. Os clientes podem modificar os pratos (por exemplo, “queijo extra”, “sem cebolas”) e aplicar cupões promocionais antes de procederem à finalização da compra. Além disso, um localizador de encomendas em tempo real apresenta o estado de preparação e entrega da refeição.

Os restaurantes recebem encomendas e tratam-nas através de um “ Dashboard do restaurante”, que apresenta os detalhes de cada encomenda, o tempo de preparação estimado e as instruções de recolha ou entrega. O sistema tem de acomodar diferentes métodos de comunicação e formatos de dispositivos - alguns restaurantes utilizam tablets ou smartphones, enquanto outros preferem sistemas POS incorporados.

O processo de checkout está integrado em vários fornecedores de serviços de pagamento, o que exige uma segurança robusta das informações pessoais e dos dados de pagamento. A *Bifanas.come* tem de cifrar informações sensíveis, seguir políticas rigorosas de retenção de dados e cumprir os regulamentos locais (por exemplo, RGPD). Uma vez que os picos de volume de encomendas ocorrem ao meio-dia e à noite, a infraestrutura do Bifanas.come deve ser dimensionada de forma eficiente - as novas encomendas e atualizações devem aparecer quase instantaneamente para evitar atrasos e suportar uma grande base de utilizadores simultâneos.

Os recursos de análise e relatórios permitem que os restaurantes participantes monitorizem as tendências de vendas e o desempenho do menu. Por sua vez, o motor de recomendação do *Bifanas.come* sugere restaurantes ou pratos aos clientes que regressam com base em encomendas anteriores. A alta disponibilidade é fundamental: quaisquer interrupções ou lentidão levam à perda de vendas e a uma má experiência do cliente. É essencial garantir que cada componente possa recuperar rapidamente de falhas - quer seja devido a um aumento da carga de utilizadores ou a uma falha de um serviço externo - para atingir objectivos de nível de serviço fiáveis.

A *Bifanas.come* também oferece uma “App de entregador parceiro”, dirigida ao entregador, que acompanha as suas tarefas e fornece otimização de rotas para garantir entregas atempadas. O estado das encomendas é atualizado para o cliente em tempo real com os dados de localização do entregador. Esta interação entre o utilizador, o restaurante e os serviços de entrega tem de ser orquestrada sem problemas, com ênfase em interfaces robustas, processamento eficiente de eventos e equilíbrio de carga para manter o desempenho durante as horas de ponta.

## Explicador

Embora a principal utilização da UML não seja para documentar a arquitetura do software, é possível usar os seus diagramas para o fazer. Clements *et al* discutem formas de usar a UML na documentação de arquiteturas[[2]](#footnote-1).

[Principais vistas](https://learning.oreilly.com/library/view/documenting-software-architectures/9780132488617/app02.html):

**1/ Module View**

Vistas de módulos mostram estruturas de arquitetura onde os elementos são unidades de implementação, ou módulos. Os módulos devem ser representados em UML como pacotes (ou classes ou interfaces, dependendo do nível de detalhe). Nas aulas TP, chamámos vistas lógicas que podem ser representadas em Diagramas de Pacotes.

**2/ Component-and-Connector View**

Os componentes C&C devem ser representados utilizando componentes UML num diagrama de componentes. Os serviços (a.k.a. interfaces) providos/consumidos são evidenciados e nomeados.

**3/ Allocation View**

As vistas de afetação apresentam mapeamentos entre elementos de software (de módulos ou vistas C&C) e elementos ambientais. Os elementos ambientais são elementos não-software (tais como nós de hardware) que estão de alguma forma associados aos elementos de software do sistema que está a ser concebido. Inclui a vista de instalação: os elementos ambientais de uma visão de instalação são elementos de hardware, tais como processadores, memória, e elementos de rede. Estes elementos podem ser representados em diagramas de instalação UML, utilizando nós.

## Como aplicar no projeto?

No projeto, o grupo deve fazer o estudo de requisitos com impacto na arquitetura e justificar a sua relevância.

As principais decisões para a conceção da arquitetura devem ser documentadas e fundamentadas.

Para além disso, o grupo deve criar um protótipo (num ambiente de prototipagem rápida) representativos das iterações a suportar e coerente com as decisões de arquitetura.

1. Exemplos de medidas relativas à conceção técnica: arquitetura de micro serviços, *load balancers*, canais seguros, tecnologia web responsiva, desenvolvimento móvel multiplataforma, *mobile-first*, *message brokers*,...) [↑](#footnote-ref-0)
2. Cf. [Anexo A](https://learning.oreilly.com/library/view/documenting-software-architectures/9780132488617/app02.html) , In: “[Documenting Software Architectures: Views and Beyond, Second Edition](https://learning.oreilly.com/library/view/documenting-software-architectures/9780132488617/)”. [↑](#footnote-ref-1)