41951 Análise de Sistemas - 2024/25 [ASis25]

Lab 10 - Documentação de interações

May 5, 2025 | Pensar e visualizar o desenrolar da colaboração entre partes do sistema.

## Percurso de aprendizagem

Em grupo.

### Colaboração entre componentes de uma arquitetura

Considere que a sua equipa de desenvolvimento está a construir um sistema **VoipGest** para a gestão de dispositivos de VoIP, vendidos a clientes empresariais.

Quando é vendida uma instalação do VoipGest é necessário registar operadores autorizados, ou seja, associar utilizadores das empresas clientes à “conta” dessa organização, para poderem monitorizar os equipamentos:

1. O Responsável da Distribuição (RespDist) pode criar utilizadores para as empresas parceiras. Para isso, o RespDist acede à webapp (fontrend do VoipGest) e indica o email do novo utilizador e a respectiva empresa.
2. O portal (=webapp) solicita a inserção do utilizador, com esses parâmetros, à API (do *backend* do VoipGest), que verifica se esse utilizador já existe associado àquela empresa na Base de dados de Operações (BD-Ops).
3. Se o utilizador pretendido for desconhecido, a API deve registar o utilizador (criar credenciais) no Sistema de Autenticação (na Cloud) que, por sua vez, solicita o Serviço de Email para enviar uma notificação.
4. Com o utilizador registado no Sistema de Autenticação (ou porque já existia ou porque acabou de ser inserido), então é adicionado à organização, na BD Operações, como um operador autorizado.
5. O Portal deve receber a indicação de que a nova associação foi bem sucedida.

Considere a discussão anterior e:

* Identifique possíveis elementos da arquitetura pretendida (“módulos” da solução).
* Visualize, num diagrama, a colaboração entre vários “módulos”, que se depreende do texto. Nota: o foco é mostrar como é que os módulos de software colaboram, não a interação com o utilizador.

### Colaboração orientada a serviços

Qualquer *marketplace* digital integra tipicamente a utilização de serviços externos, por exemplo, para articular com fornecedores ou transportadoras (B2B), ou delegar os pagamentos numa plataforma de micropagamentos.

Neste caso, podemos dizer que há a articulação com “serviços” digitais de terceiros usando frequentemente [API REST](https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design).

* Tendo em conta os requisitos típicos dos *eMarketplaces*, e, se aplicável, do seu projeto de grupo, considere um dos serviços externos que podia ser potencialmente integrado e procure obter informação específica de APi que poderiam ser usadas (e.g.: API para integrar com o PayPal a partir do frontend para concluir um pagamento; API para integrar com a DHL e colocar um novo pedido de entrega; etc.)
* Especifique o cenário de colaboração com a API externa, incluindo também um diagrama de sequência.

Nota: assegure-se que o resultado é relevante para programadores, reflete a utilização da API de forma verossímil e usa UML bem formada, especialmente se obteve informação de AI generativa.

### Colaboração entre objetos (de código)

Considere a implementação existente de uma hipotética aplicação que gere pedidos de um restaurante. Um output ilustrativo está exemplificado a seguir.

Para efeitos de demonstração, o programa gera uma ementa aleatória, com alguns pratos adicionados e, depois, simula um pedido, escolhendo dois pratos dessa ementa.

O respectivo projeto de implementação (Java) está [disponível online](https://github.com/icoUAv/maslabs); ponto de arranque DemoClass.java 🡪 main(). Poderá [transferir o projeto](https://github.com/icoUAv/maslabs/archive/refs/heads/master.zip) e abrir localmente, mas não é estritamente necessário

* Comece por criar uma visualização da estrutura do código (pasta src/ementas/\*). Represente a ocorrência de classes; atributos; associações entre classes (uma classe usa atributos cujo tipo de dados é outra classe do modelo, simples ou como uma colecção): situações de especialização; operações oferecidas pelos objetos de cada classe[[1]](#footnote-0).
* Analisando o código disponível, visualize as interações entre objetos que ocorrem quando é solicitada a operação: Pedido 🡪 **calcularCalorias**();

Para isso, “siga” o fluxo de execução, veja os objetos e operações envolvidas, e recorra a um diagrama de sequência. Para criar cada lifeline, pode arrastar a classe correspondente (Pedido,…) da árvore do modelo para o diagrama, caso já as tenha criado.

**Tabela 1: *output* do programa principal, simulando um pedido de comida.**

| **A preparar os dados...**  A gerar .. Prato [nome=**Dieta n.1**,0 ingredientes, preco 200.0]  Ingrediente 1 adicionado: Cereal [nome=Milho; Alimento [proteinas=19.3, calorias=32.4, peso=110.0]]  Ingrediente 2 adicionado: Peixe [tipo=CONGELADO; Alimento [proteinas=31.3, calorias=25.3, peso=200.0]]  A gerar .. Prato [nome=**Combinado n.2**,0 ingredientes, preco 100.0]  Ingrediente 1 adicionado: Peixe [tipo=CONGELADO; Alimento [proteinas=31.3, calorias=25.3, peso=200.0]]  Ingrediente 2 adicionado: Legume [nome=Couve Flor; Alimento [proteinas=21.3, calorias=22.4, peso=150.0]]  A gerar .. Prato [nome=**Vegetariano n.3**,0 ingredientes, preco 120.0]  Ingrediente 1 adicionado: Cereal [nome=Milho; Alimento [proteinas=19.3, calorias=32.4, peso=110.0]]  Ingrediente 2 adicionado: Cereal [nome=Milho; Alimento [proteinas=19.3, calorias=32.4, peso=110.0]]  A gerar .. Prato [nome=**Combinado n.4**,0 ingredientes, preco 100.0]  Ingrediente 1 adicionado: Cereal [nome=Milho; Alimento [proteinas=19.3, calorias=32.4, peso=110.0]]  Ingrediente 2 adicionado: Cereal [nome=Milho; Alimento [proteinas=19.3, calorias=32.4, peso=110.0]]  **Ementa para hoje:** Ementa [nome=Menu Primavera, local=Loja 1, dia 2020-11-22T21:08:45.624777300]  Dieta n.1 200.0  Combinado n.2 100.0  Vegetariano n.3 120.0  Combinado n.4 100.0  **Pedido gerado:**  Pedido: Cliente = Joao Pinto  prato: Prato [nome=Combinado n.2,2 ingredientes, preco 100.0]  prato: Prato [nome=Combinado n.2,2 ingredientes, preco 100.0]  datahora=2020-11-22T21:08:45.813778700]  Custo do Pedido: 200.0  Calorias do Pedido: 95.4 |
| --- |

## Como aplicar no projeto?

Nem todos os projetos terão o mesmo potencial para usar modelos de interação com diagramas de sequência.

Em todo o caso, é expectável que os projetos modela alguma destas situações:

* O sistema desenvolvido precisa de usar sistemas externos, tal como se depreende da existência de certos atores (que seriam sistemas externos). A forma de integração com esses sistemas deve ser concretizada.
* O sistema desenvolvido deve articular com fornecedores num cenário B2B. Como é que essa interação deve acontecer? Quem inicia? Em que casos?
* Existem sistemas ciberfísicos que serão integrados e para os quais há um protocolo de comunicação. Como é que é iniciada a ligação? Qual o desenrolar do protocolo?

1. Pode-se ignorar os métodos triviais que, até certo ponto, só prejudicam a legibilidade do diagrama: getter e setter; equals(), hash(). [↑](#footnote-ref-0)