**Guilherme Matos (114252), Gustavo Bastos (114034), Gustavo Garcia (114411), André Brito (104119)**

Grupo 402, v2024-02-23

RELATÓRIO

Lab 4: Modelação de comportamento (interações)

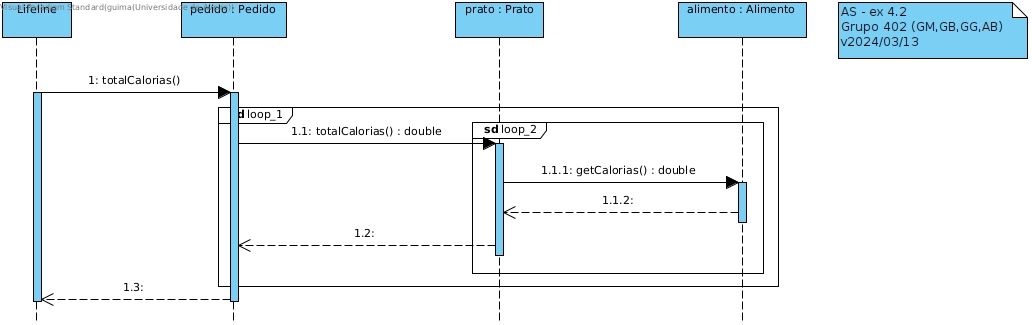
## Exercício 4.1

O user inicia uma ação de pagamento. A aplicação IOS faz um pedido de pagamento ao PKVC (PassKit View Controller), que responde com um pedido de aprovação ao user. O user autoriza o pagamento e o PLVC retorna a informação do mesmo.

É enviado um Apple Token do PKVC para o Apple Server, que irá ser encriptado com o Merchant ID pelo mesmo, e retornado (encriptado) ao PKVC. O Token encriptado é então enviado do PKVC para a aplicação IOS.

É efetuado um pedido de pagamento (CreditCardAuthRequest) por parte da aplicação IOS para o serviço de pagamento. É então feito um Transform Request e, de seguida, é feita a decriptação dos dados do Token. O serviço de pagamento efetua um request de Gateway Payment ao Gateway, e obtém uma resposta. É feita uma transformação da resposta, e, por fim, a resposta de pagamento (CreditCardAuthReply) chega à aplicação IOS.

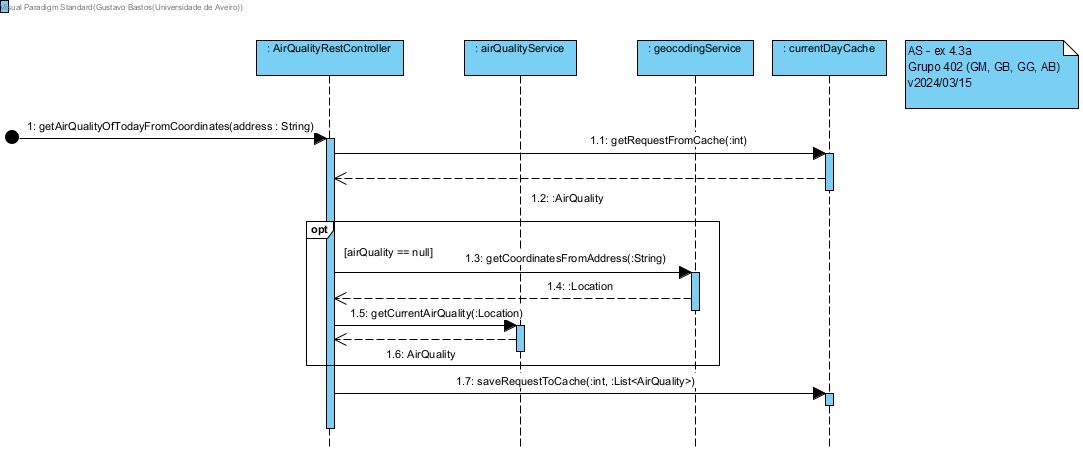
## Exercício 4.2

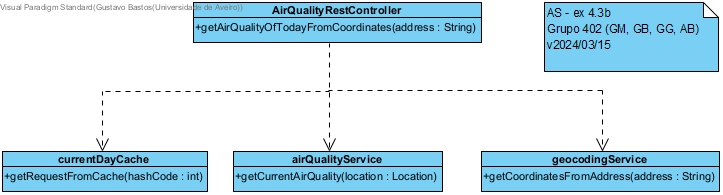


* O indivíduo ou ator interessado procura saber o total de calorias ( totalCalorias() ) num determinado pedido.
* Para determinar o total de calorias de um pedido é necessário percorrer todos os pratos do mesmo e portanto existe um loop que os vai percorrer, procurando saber o total de calorias de cada prato.
* Para determinar o total de calorias de um determinado prato é ainda preciso ir a cada alimento dentro do mesmo, e portanto existe outro loop que o vai fazer. Tendo os valores calóricos de cada alimento é então tido o total de calorias de cada prato.
* Tendo o total de calorias de cada prato é então tido o total de calorias de um pedido.
* É então enviado o valor do total de calorias a quem fez o pedido inicial.

## Exercício 4.3 a)

A classe AirQualityRestController tem o método getAirQualityOfTodayFromCoordinates() que quando chamado interage com outras classes da seguinte forma:

* Começa por chamar o método getRequestFromCache() que faz parte da classe currentDayCache e que devolve um objeto do tipo AirQuality;
* Seguidamente, caso se verifique que a variável local airQuality é null, o método getCoordinatesFromAddress() da classe geocodingService é chamado; Este devolve um objeto do tipo Location;
* É também chamado o método getCurrentAirQuality() da classe airQualityService que devolve um objeto do tipo AirQuality;
* Finalmente é chamado o método saveRequestToCache() da classe currentDayCache que não retorna nada;

**Exercício 4.3 b)**

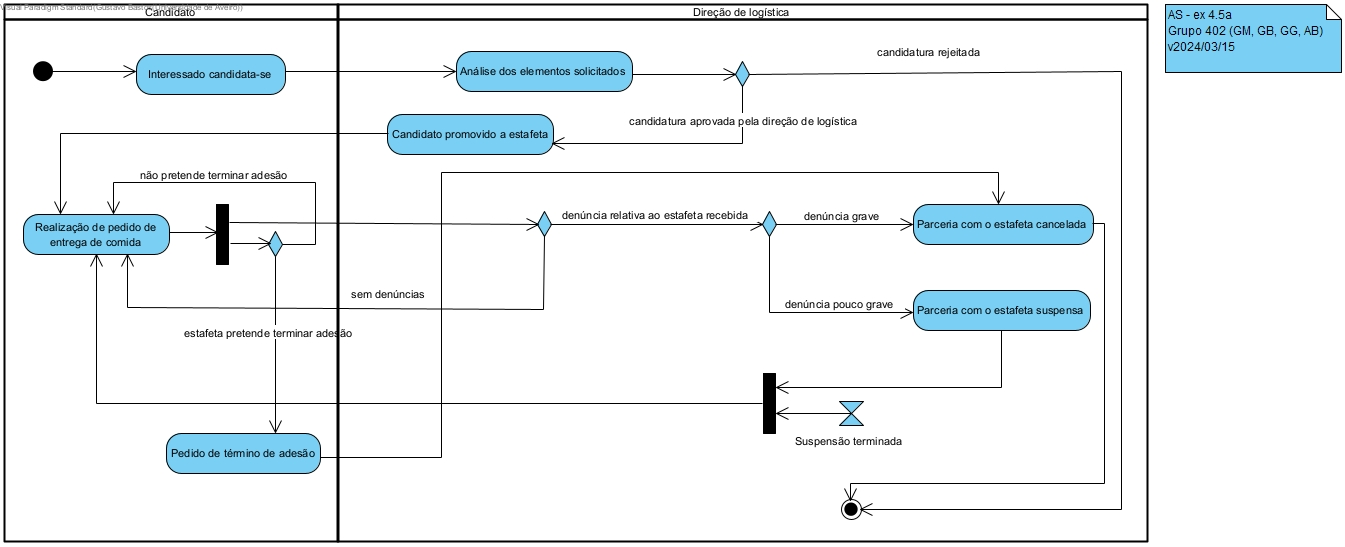
## Exerc**íc**io 4.4

## 

* Novo: Estado inicial quando um resultado é inserido pela primeira vez na cache.
* Válido: Estado após um put de um novo resultado ou atualização de um existente.
* Dirty: Estado de um resultado expirado que ainda não foi removido da cache.
* Periodicamente, a cache é reavaliada.
* Miss: Estado quando um get é feito para um resultado que não existe na cache.
* Removido: Estado após um resultado expirado ser removido da cache.

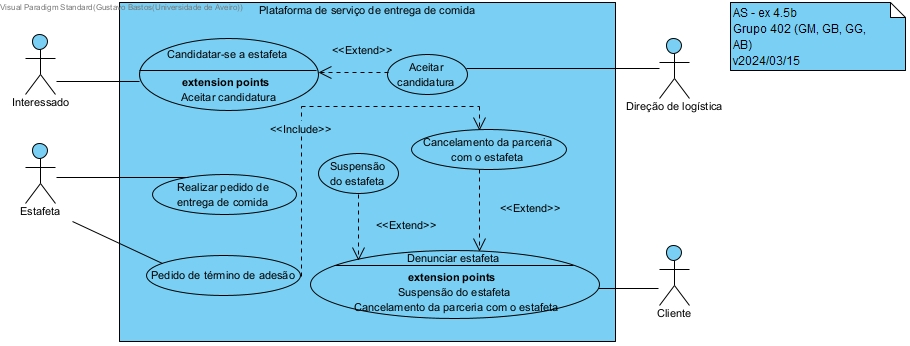
## Exerc**íc**io 4.5

**a)**



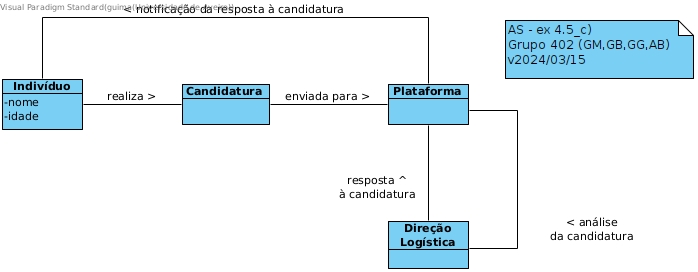
* Inicialmente alguém interessado candidata-se à posição de estafeta.
* Caso a sua candidatura seja aceite, este torna-se estafeta
* À medida que vai fazendo entregas pode receber denúncias dos clientes
* Denúncias graves equivalem ao término da parceria com o estafeta
* Denúncias menos graves podem apenas dar lugar a uma suspensão, que assim que terminada, permitirá ao estafeta continuar a fazer entregas
* Eventualmente o estafeta pode pedir o cancelamento da parceria

**b)**



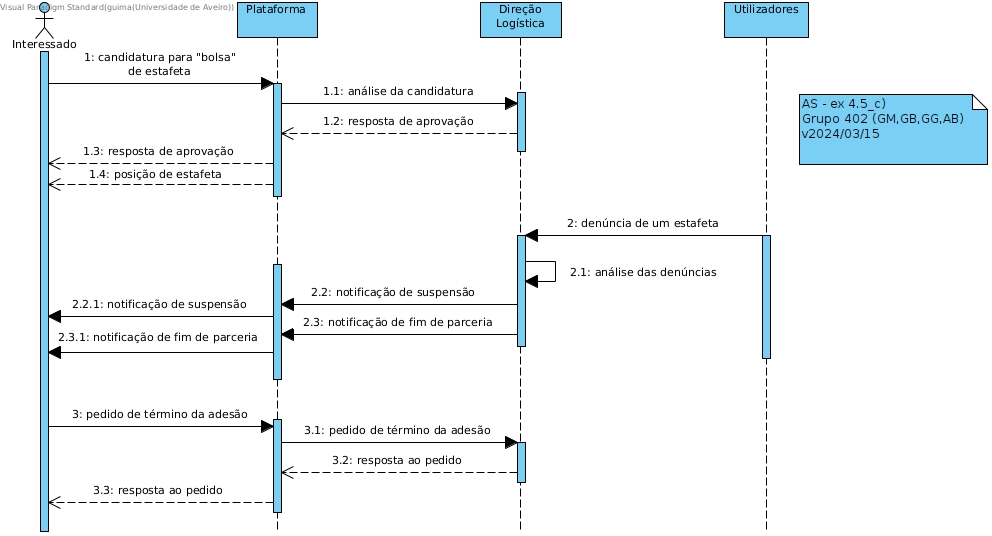
## **Exercício 4.5**

**c)**



A representação deste caso hipotético num diagrama de classes não parece ter tanta relevância como outras representações. No entanto, mostra as relações entre cada classe.

**d)**

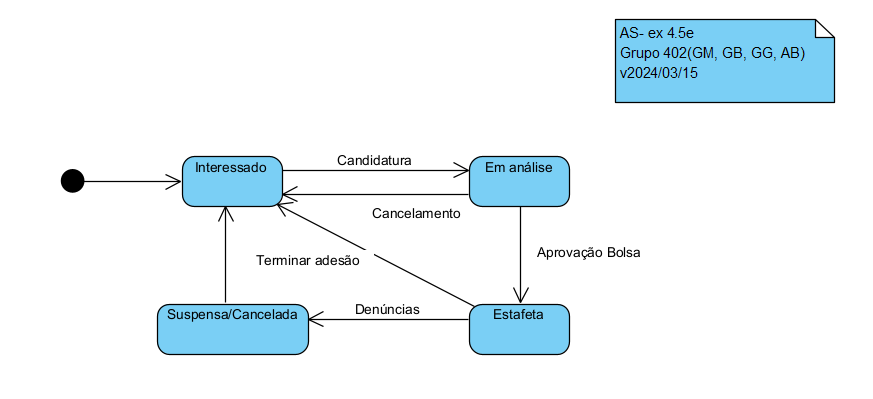


**Exercício 4.5**

**d) (continuação)**

* Representar o caso hipotético num diagrama de sequência já tem um grau de relevância maior pelo que podemos verificar os acontecimentos e a sua sequência de uma forma simples.
* Conseguimos ver as interações entre os atores e as classes de forma bastante clara e perceber de melhor forma como o sistema funciona entre si.

**e)**



Criar uma máquina de estados fornece uma representação visual e estruturada do processo de gestão da relação com os estafetas parceiros. Isso pode ajudar a identificar e compreender melhor as diferentes etapas do processo e as transições entre esses estados.