

## Lab. 5: Activity Diagrams

↳ Exercício 1: (b) e (c)

↳ Exercício 2: (a) ✓ (b) ✓ (c) ✓ (d) F (e) F (f) F

↳ Exercício 3: • o diagrama tem um deadlock logo no início. ( ) nó de decisão entre os estados de ação "Inspect forms" e "Display ..."

• a atividade pode acabar chegando a um fim graças ao estado de atividade "Apply to University". Mesmo que seja uma atividade em si, é necessário saber o que se decorre após essa atividade chegar ao fim

• a atividade pode nunca acabar pois depois do fork se no nó de decisão que vem a seguir ao estado de ação "Search for application system" caso a escolha seja "No matches" então o 1º segundo fluxo dá merge com o primeiro. De tal forma nunca será possível fazer join dos dois fluxos

• existe um nó de decisão guardado

• o join pode chegar a não acontecer devido ao nó de decisão a seguir a "Display ...", pois o segundo fluxo pode acabar a fazer merge com o primeiro

↳ Exercício 4: • o segundo estado de ação do fluxo pode "separar-se" em dois fluxos, dependendo de uma escolha. Deveria ser um nó de decisão

• acontece o mesmo depois do estado de ação "check Applicants list" e "Search for applicant" e "Display list for Rother"

• o join precisa que os dois fluxos, mas um deles pode acabar antes de chegar ao join

• o estado de ação "Em ... n" não se pode encontrar entre duas swimlanes

↳ Exercício 5: • não faz sentido haver uma swimlane que apenas tem uma condição de guarda

• um estado de ação não pode ter uma condição de guarda

• um estado de ação não se pode dividir em dois fluxos sem um fork

• depois da execução do sinal "Possible security Risk" nada é dito do que acontece a seguir

↳ Exercício : o extension point encontraria-se aconteceria antes do ponto 7 do use case : Extension point: Create Account

Use Cases: Create Account

Desc: the client decides to create an acc in the system

Actors: Main: client - Secondary: none

Pre-conditions of req 1: the client doesn't have an acc already

Main flow:

1. the client inserts their information
2. the system saves the client's data
3. returns to step 1 if

voltam ao passo 2  
ou não pode ser?

Post-conditions: the client's data is now saved in the system  
the system has a new account

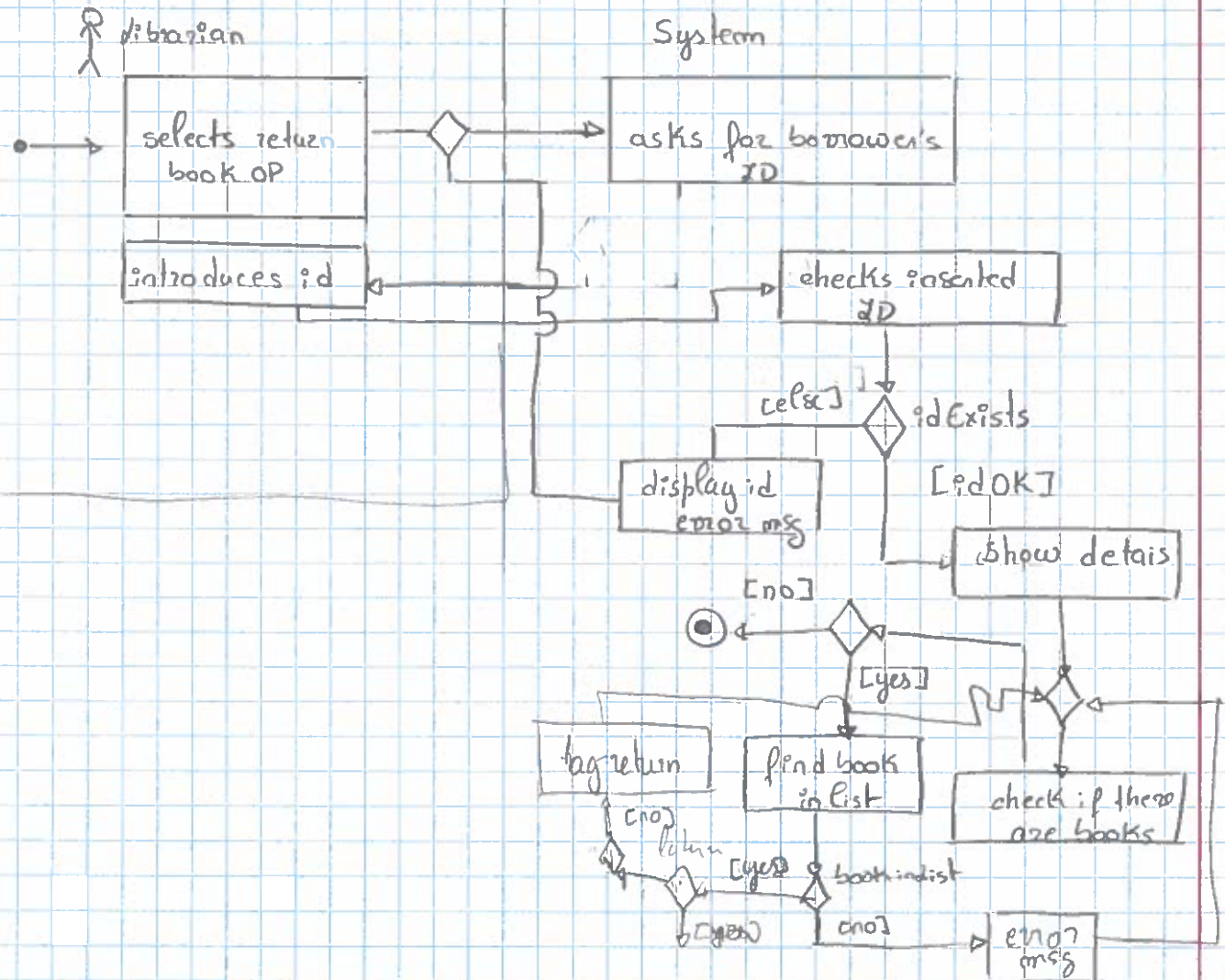
### ↳ Exercício 2: B

↳ Exercício 9: 4

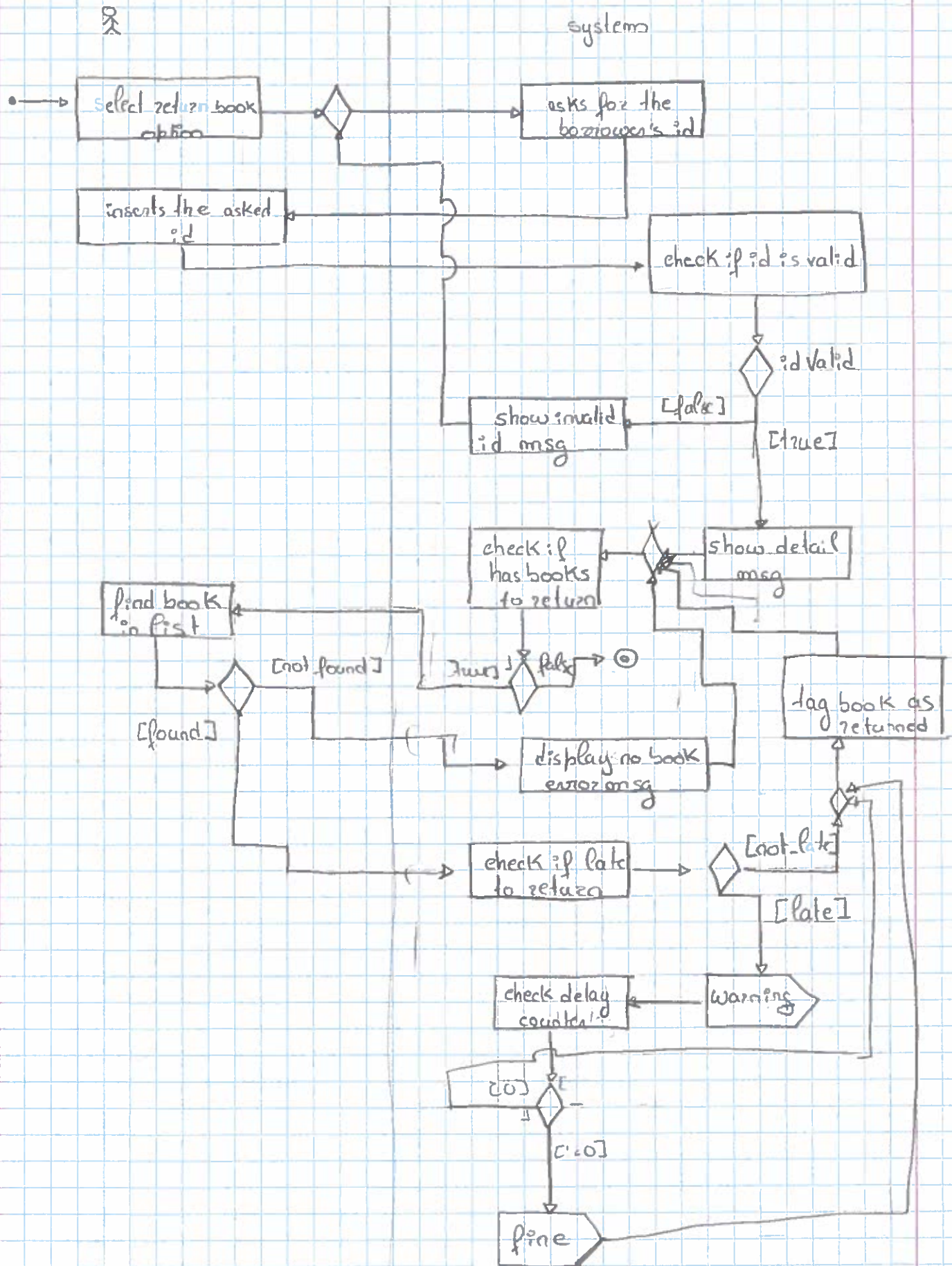
#### ↳ Exercício 10: E

### ↳ Exercício 1:

## "Hand Back Book"

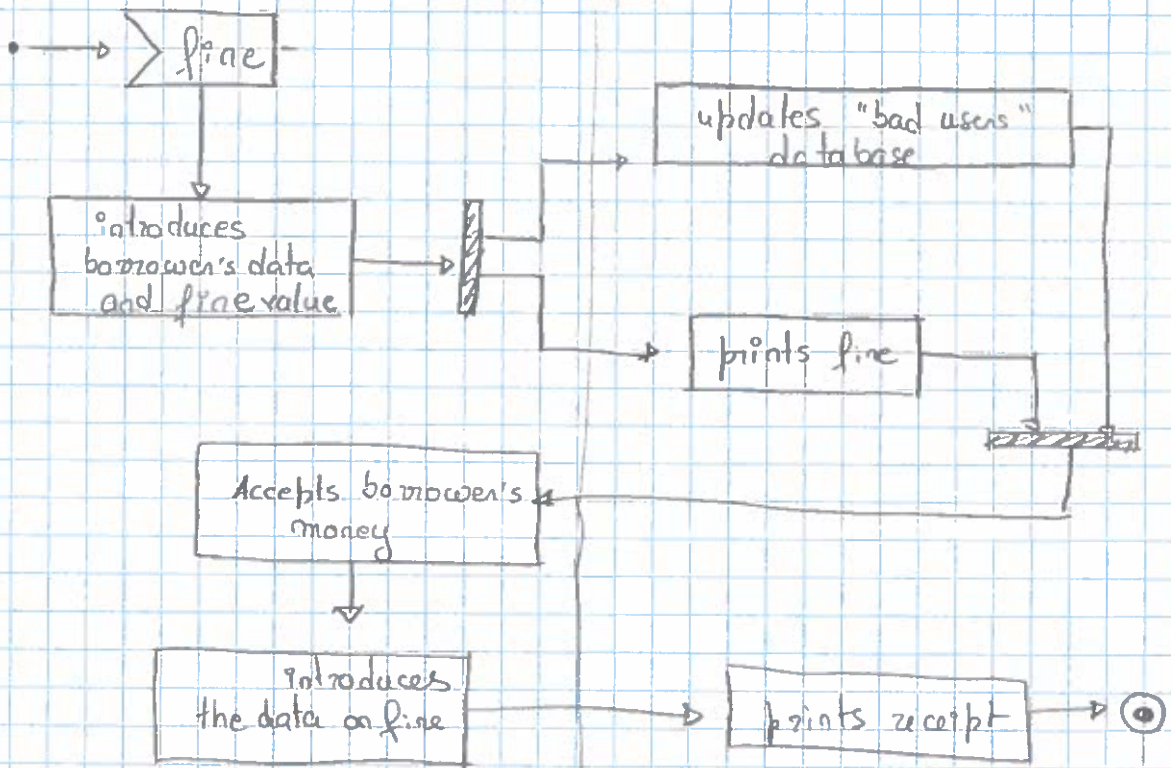
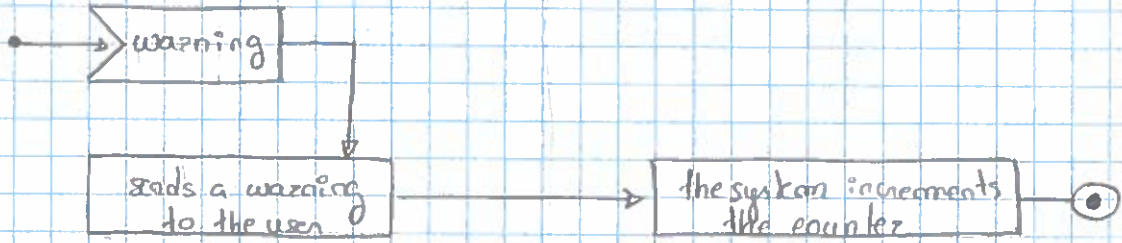


# ↳ Exercício 6:



librarian

system



# Exercício 11:

actor: driver

arrives at the car parking

system

checks if there are places available

[no places]

show message saying there's no availability

show 3 options

select one of the shown ops

[ticket]

[uv]

[fu]

collect ticket

print ticket

check if valid

iss

check if valid

[no]

show error

collect back printed ticket

[not valid]

entrance is saved

barrier is lifted

detect passage

put barriers down



## → Activity Diagrams (Aula Zucca)

1/ (c)

2/ (a) ✓

(b) ✓

(c) ✗

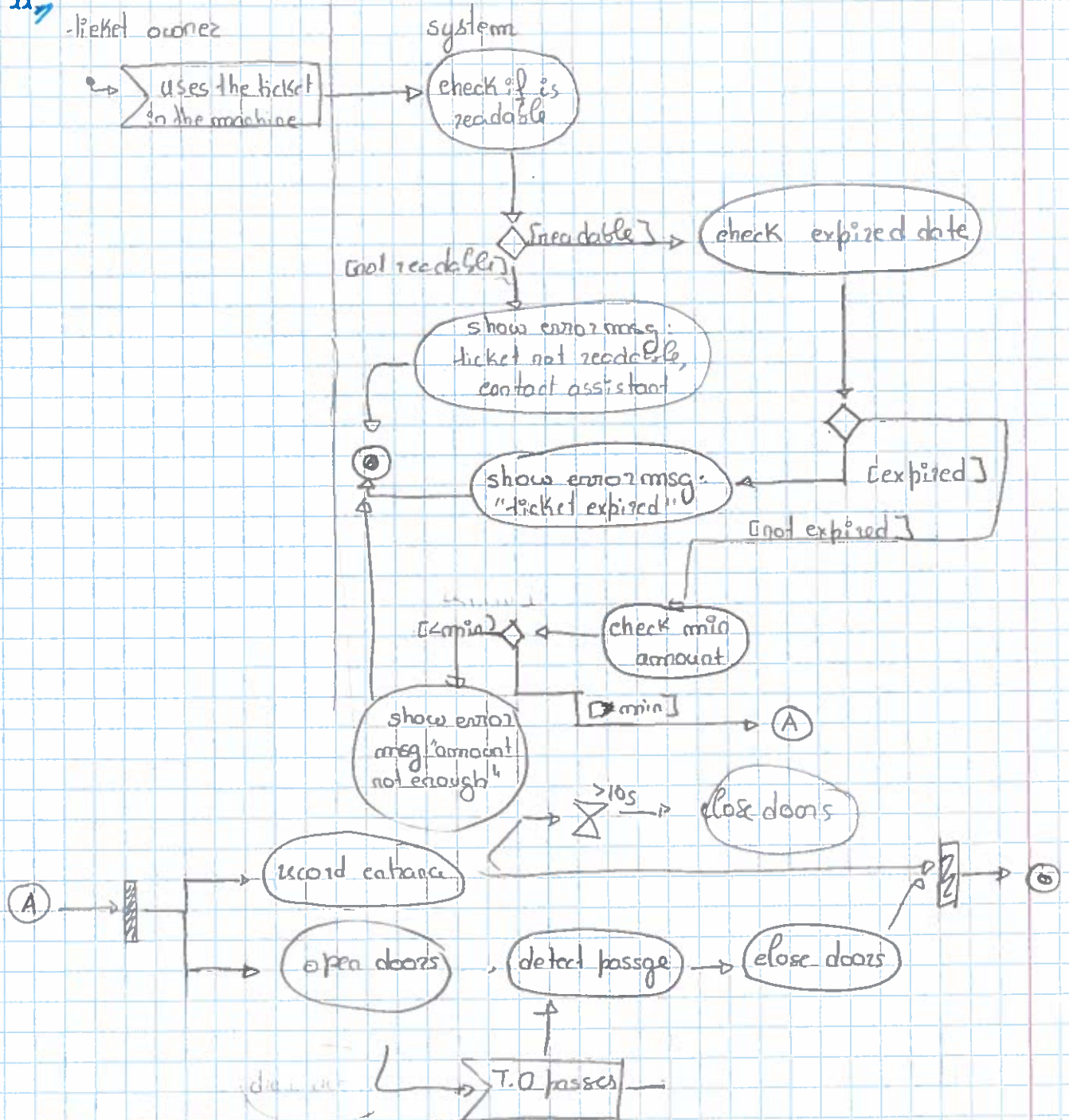
(d) ✗

(e) ✗

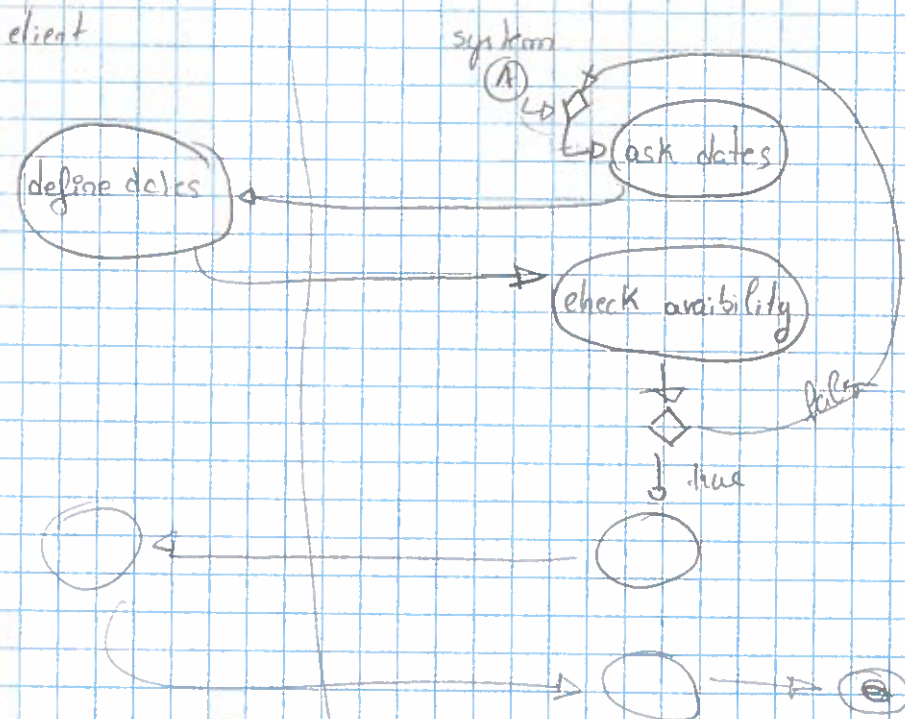
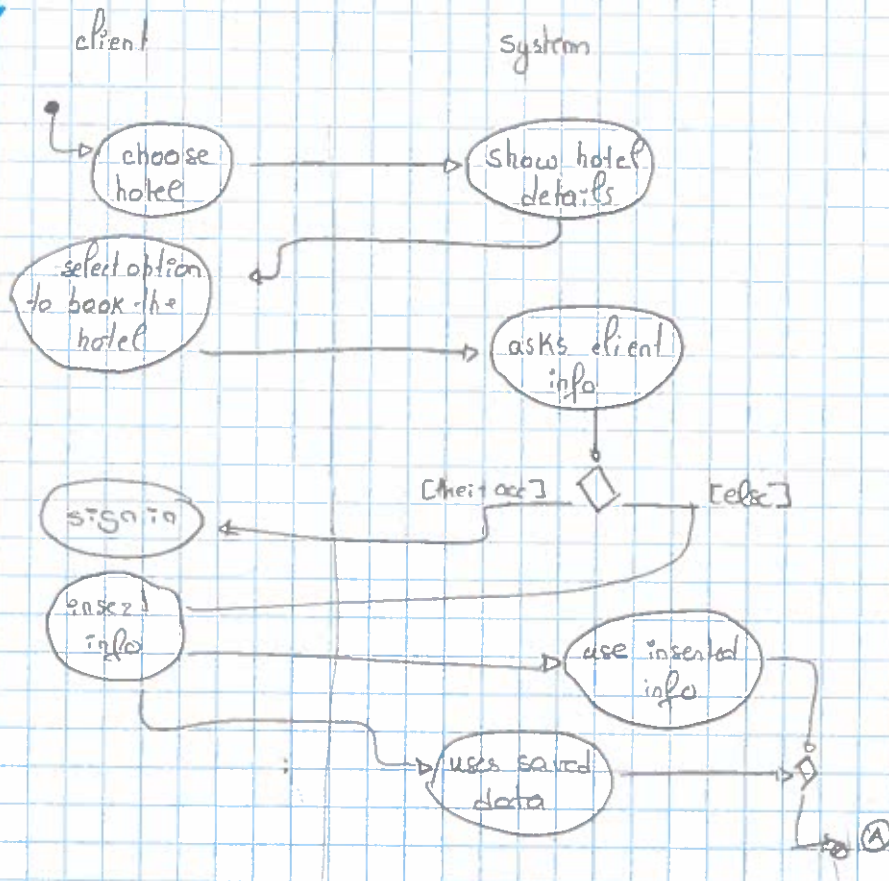
(f) ✗

3/ - - -

11/

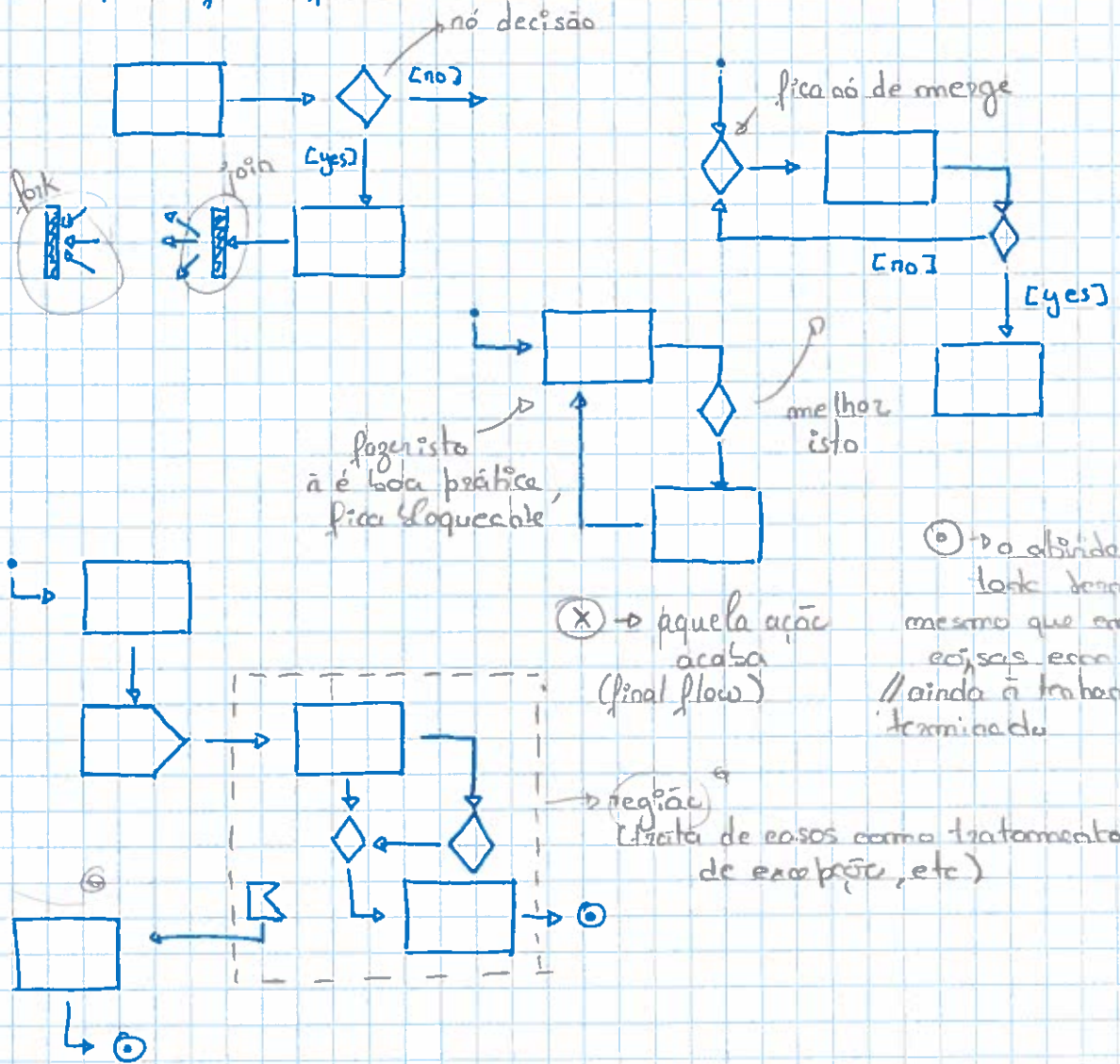


7



## → Activity Diagrams 2017/2018

→ serve para definir o fluxo



1/ (b) e (c)

2/ (a) ✓ (b) ✓ (c) ✓ (d) F (e) F (f) ✓

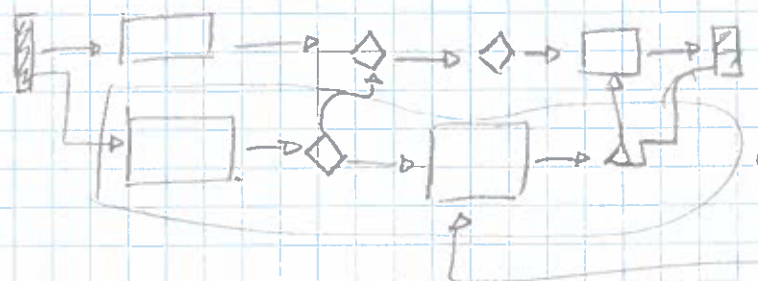
3/

• deadlock no diagrama →

• quando vai para a atividade não se diz o que acontece a seguir

Apply to UML

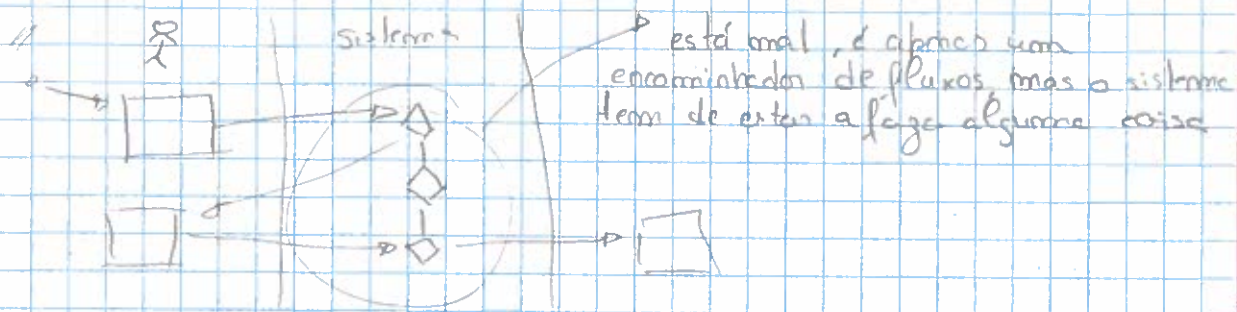
• nó de decisão sem guardas



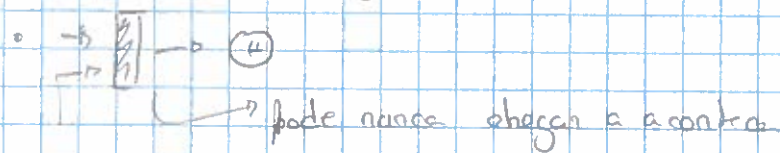
o fluxo pode nunca chegar a este ponto (logo nunca chega ao B)

4

5 a swimlane "forms not filled out" não flag erro e nem tem sequer nenhuma atividade



• aqui acontece a seguir ao sinal [ ] , não pode ficar pendurado



8 B

9 A

6

