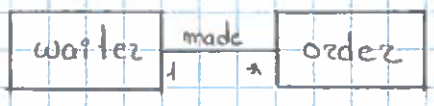
Prática (05): Class Diagram

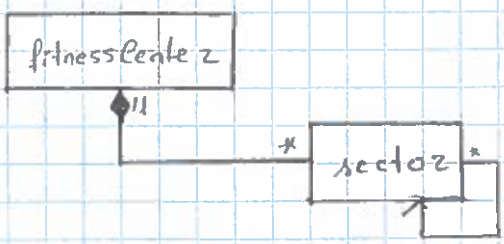
1. (a) T (b) F (c) F (d) T (e) F (f) F (g) T

2. (a) T (b) F (c) T (d) T

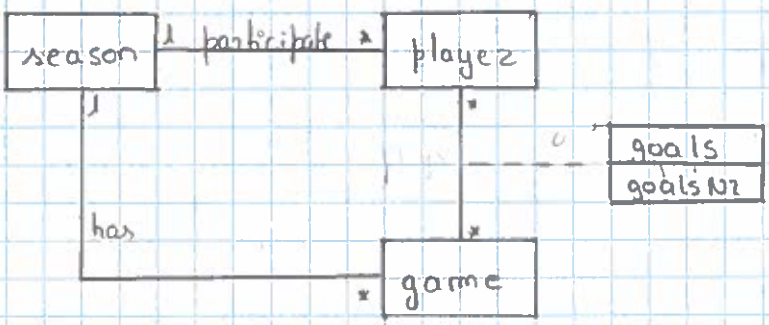
3.



4.



5.

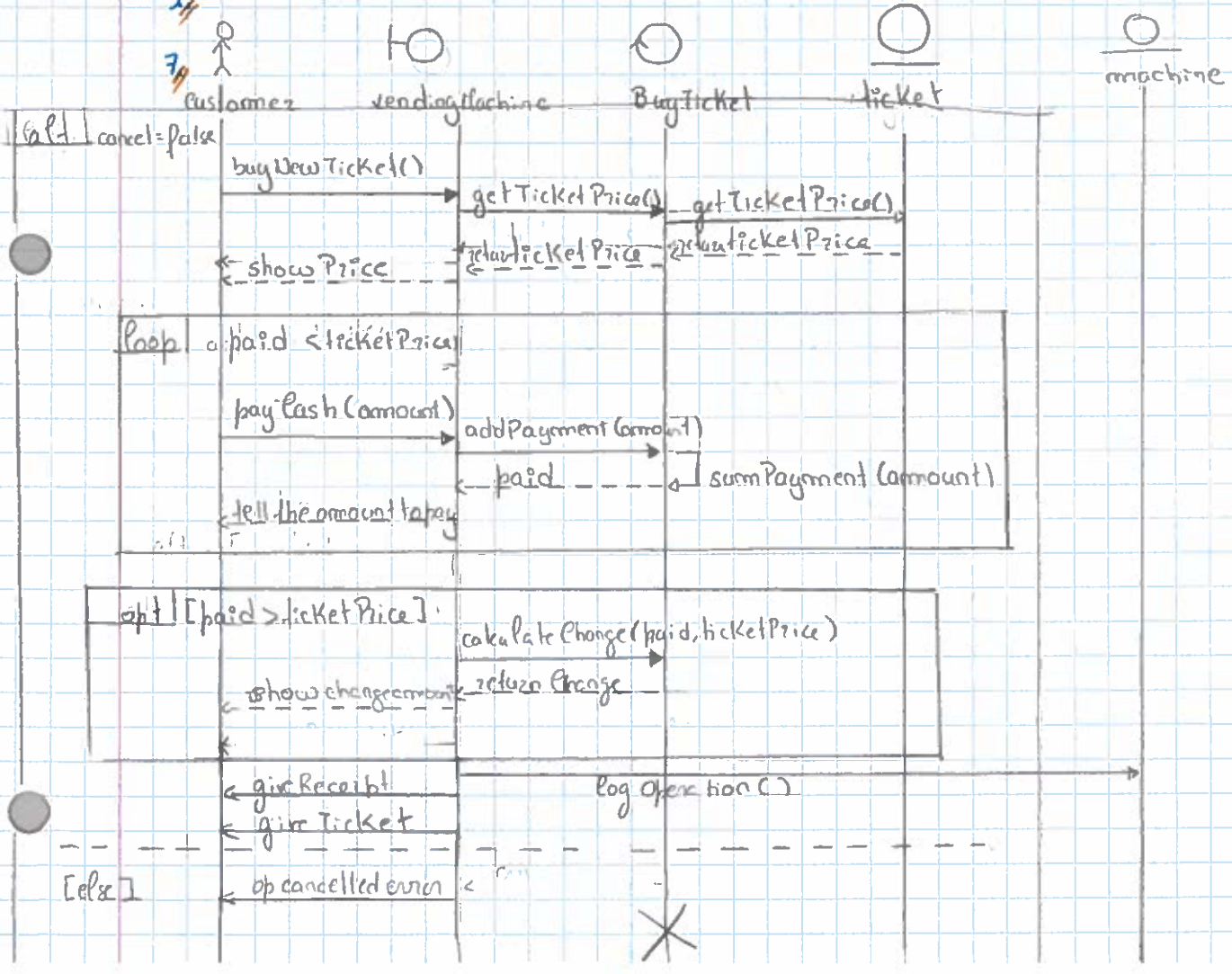


6.

Prática (06): Interaction Diagrams

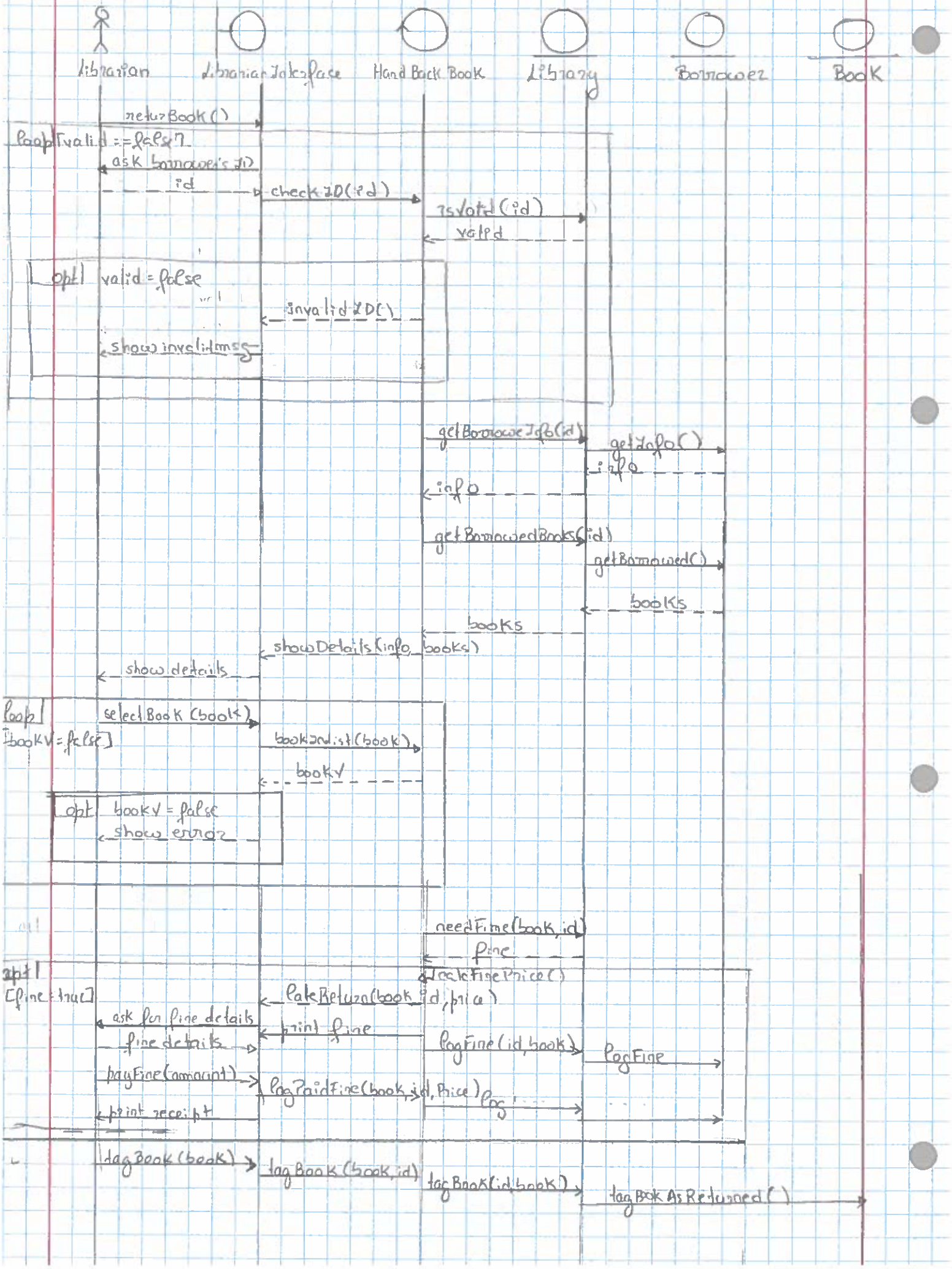
1.

7.



9

• Hand Back Book



Prática (07): OCL Class Diagram

1) - Airport Model

class Flight
attributes
flightNumber: Integer
cost: Real
operations
getFlightNumber(): Integer

class IndirectFlight < Flight
attributes
levy: Real
end

class Airline
attributes
profitRate: Real
name: String
end

class Airport
attributes

class Passenger
attributes
name: String
end

class Time
attributes
hour: Integer
minute: Integer
seconds: Integer
month: Integer
day: Integer
year: Integer
end

class Aircraft
attributes
maker: String
model: String
capacity: Integer
end

operations
getCapacity(): Integ

- associations

association owns between
Airline [1] line
Flight [0..*] flights
end

association goes between
Airport [1] destination attributes:
Flight [0..*] flights landingTime: Time
end

association boards between
Passenger [1..*] passengers
Airport [1] airport
attributes gate: Integer end

association travels between
Passenger [1..*] passengers
Flight [1] flight
end

association stop between
IndirectFlight [1..^] flights
Airport [0..*] transit_airport
attributes: landingTime: Time end
departureTime: Time)

association with between
Aircraft [1] plane
Flight [1..*] flight
end

association departures between
Flight [0..*] role outgoing
Airport [1] role departure
attributes
departureTime: Time
end

association flying between
boards [0..*] role embarks
Flight [1] role Flight
end

(c) context Flight
inv noDups: self.allInstances -> forAll (f1, f2 : Flight | f1 <> f2 implies (f1.flightNumber <> f2.flightNumber and (f1.departures.departureTime.year <> f2.departures.departureTime.year) or (f1."/"month <> f2."/".month) or (f1."/"day <> f2."/".day)))

(d) context Flight
inv cap: self.plane.capacity >= self.passengers->size()

(e)
context Flight
inv: self.destination <> self.departure

(f) contex IndirectFlight
inv: (self.transit_airport <> self.departure) and
(self.transit_airport <> self.destination)

// -- model Sudoku

class Matrix
end

class SubMatrix
end

class Number
attributes
value : Integer
end

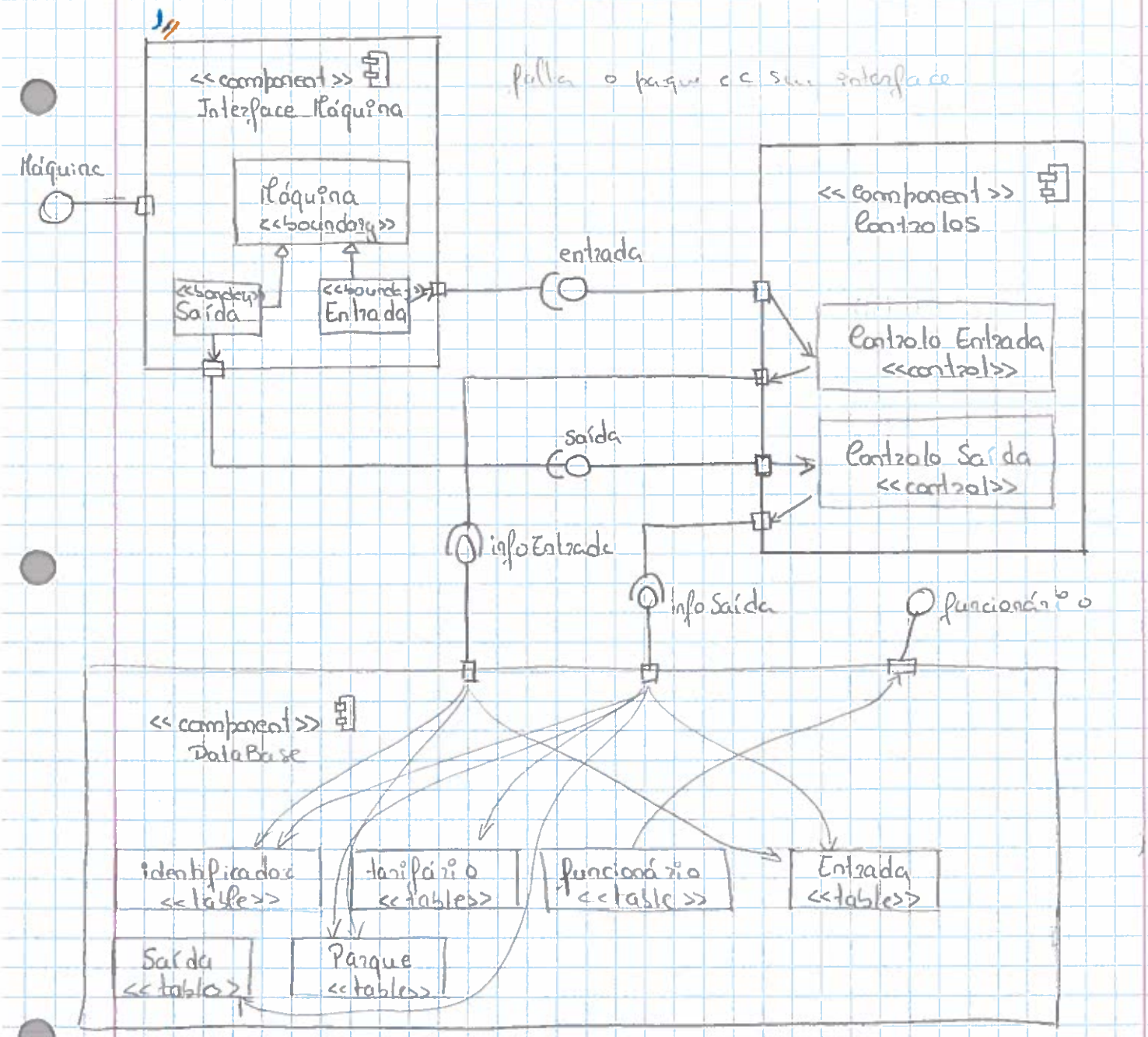
-- associations

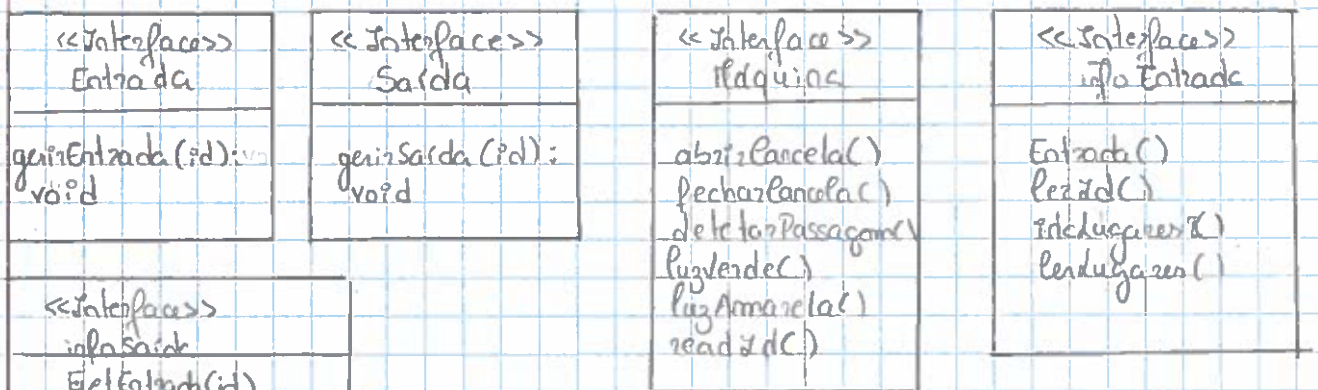
association composedBy between
Matrix [1] role game
SubMatrix [9] role subs
end

association line between
Number [3] role numbers
SubMatrix [1] role sub
attributes
indice: Integer
end

association column between
Number [3] role numbers
SubMatrix [1] role sub
attributes
indice: Integer
end

Prática (08): Component / Deployment Diagrams





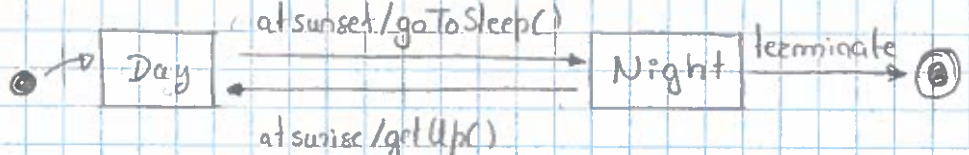
2/

→ Prática (X): State Machines

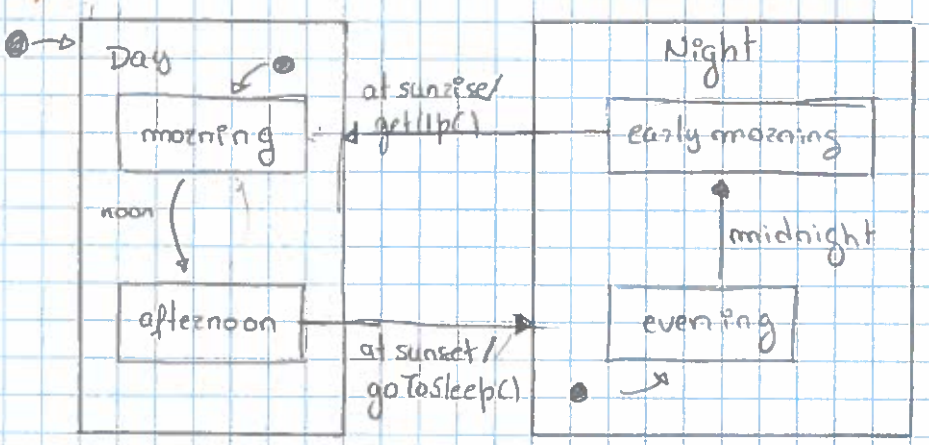
1/ (a)(ii) (b)(iv) (c)(vi) (d) X (e) (f)(i) (g)(iii)

2/ não foi dado

3/



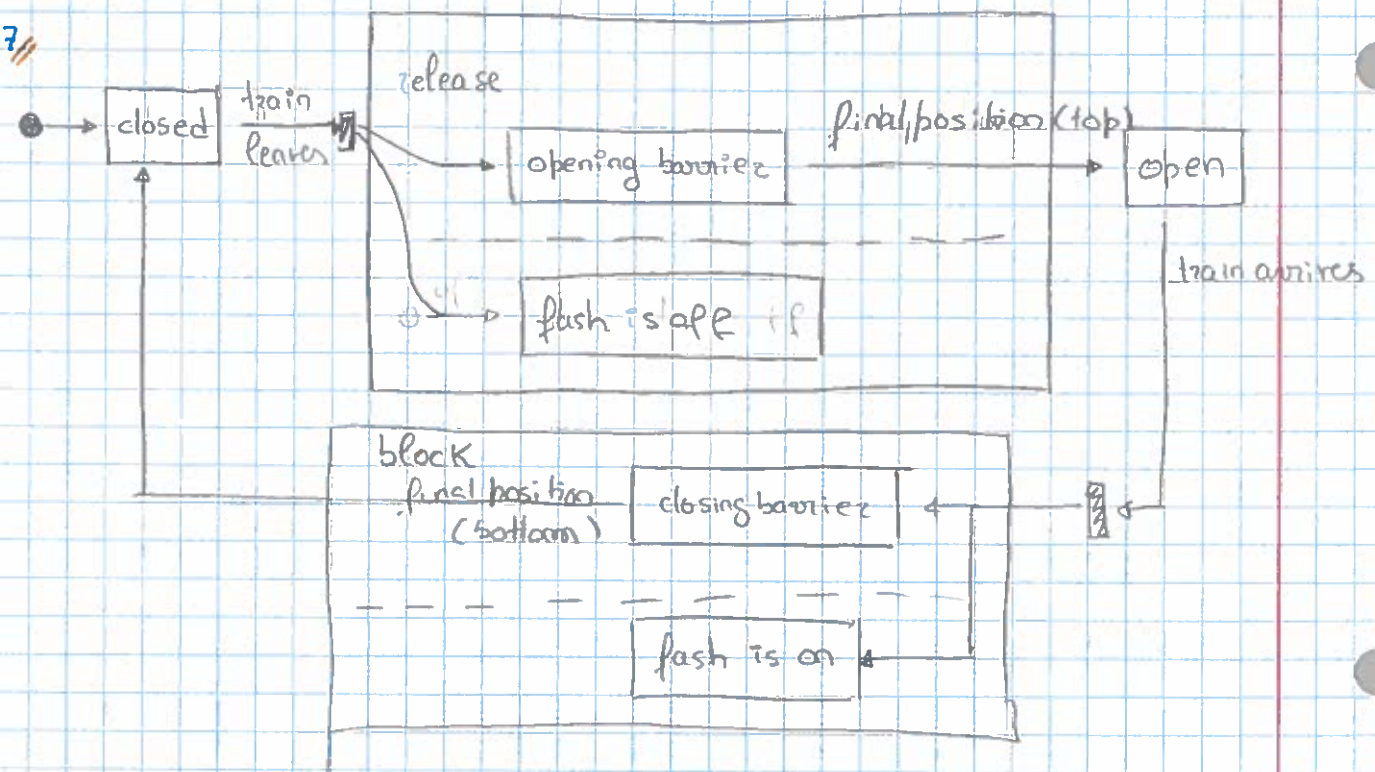
5/



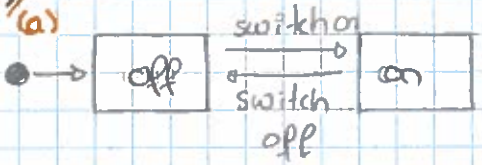
5/ — não dado (orthogonal substates)

6/ (a) iii (b) i (c) i

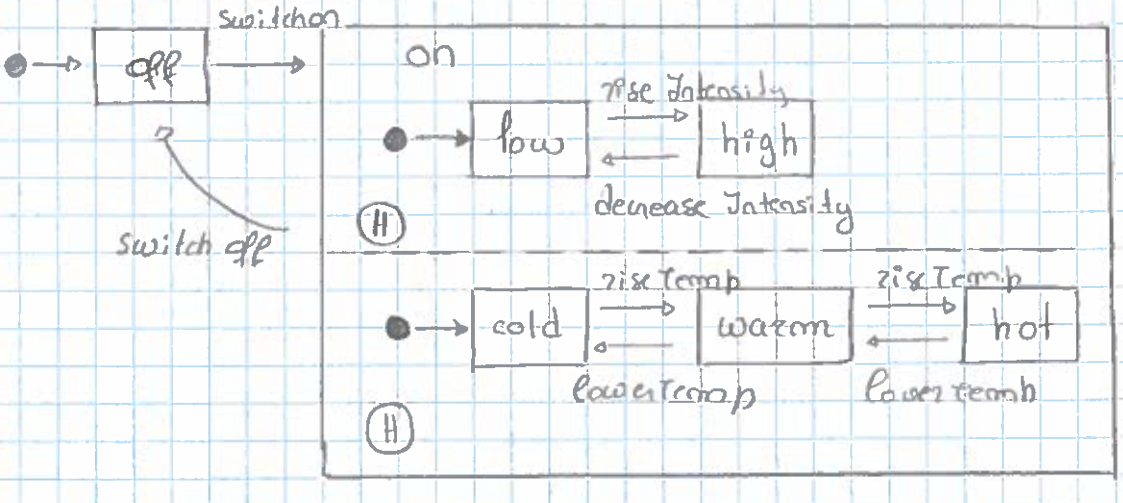
7/



9 (a)



(b)



(c) são os (H)

(d)