

# Relatório Final de Técnica de Programação 1

Felipe Luís Pinheiro\*- 18/0052667

Gabriel Teixeira da Silva†- 170079538

Paula Vychthória de Araújo Lima‡- 170112446

Rodrigo Belone Ramos§- 170071804

28 de março de 2019

## Resumo

Este relatório faz parte do processo de avaliação da disciplina Técnica de Programação 1 (117889) do Instituto de Ciências Exatas, Departamento de Ciência da Computação da Universidade de Brasília - Brasil, DF, Ministrada pelo professor Cristhian Ivan Riaño Jaimes.

Neste relatório mostramos o desenvolvimento UML de 4 projetos (Locadora de Carro, Sistema de Cinema, Sistema de Hotelaria e Locadora de Video Game) e por fim desenvolvemos o projeto ... até a sua conclusão como software completamente funcional.

## 1 Introdução

A UML tem origem na compilação das “melhores práticas de engenharia” que provaram ter sucesso na modelagem de sistemas grandes e complexos. Sucedeu aos conceitos de Booch, OMT (Rumbaugh) e OOSE (Jacobson) fundindo-os numa única linguagem de modelagem comum e largamente utilizada. A UML pretende ser a linguagem de modelagem padrão para modelar sistemas concorrentes e distribuídos.

A UML ainda não é um padrão da indústria, mas esse objetivo está a tomar forma sob os auspícios do Object Management Group (OMG). O OMG pediu informação acerca de metodologias orientadas a objetos que pudessem criar uma linguagem rigorosa de modelagem de software. Muitos líderes da indústria responderam na esperança de ajudar a criar o padrão. [1]

## 2 Projetos

A seguir mostramos os 4 projetos proposto pelo professor:

---

\*fpinheiro@gmail.com

†gabrielbsb21@outlook.com

‡araujopaula534@gmail.com

§rodrigobeloner@gmail.com

## 2.1 Locadora de Carro

Parte 1: Levantar em consideração os seguintes requisitos:

- A empresa tem muitos automóveis. Cada automóvel tem atributos como número da placa, cor, ano, tipo de combustível, número de portas, quilometragem, RENAVAM, chassi, valor de locação, etc.
- Cada carro tem um modelo e uma marca, mas o modelo pode relacionar-se a muitos carros, e uma marca pode referir-se a muitos modelos, embora cada modelo só tenha uma marca específica.
- Um carro pode ser alugado por muitos clientes, em momentos diferentes, e um cliente pode alugar muitos carros. É preciso saber quais carros estão locados ou não. Sempre que um carro for locado é preciso armazenar a data e a hora de sua locação e, quando for devolvido, a data e hora de devolução.

Parte 2: Uma locadora de carros deseja fazer um sistema para armazenar as informações das locações que os clientes fazem. A locadora possui várias agências (código agência, e localidade). É necessário registrar tanto a data, hora e agência para cada alocação como para sua devolução e dados do cliente. A locação pode ser tanto diária (precisa de dias previstos para devolução) como por período (aplica porcentagem de desconto dado no valor da diária). A locadora armazena dados dos carros (modelo, placa, cor, ano e data de aquisição) e classifica em uma categoria para definir seu valor de diária de locação

## 2.2 Sistema de Cinema

Requisitos gerais:

- Um cinema pode ter muitas salas, sendo necessário, por tanto, registrar informações a respeito de cada uma, como sua capacidade, ou seja, o número de assentos disponíveis.
- O cinema apresenta muitos filmes. Um filme tem informações, título e duração. Assim, sempre que um filme for ser apresentado, deve-se registrá-lo também.
- Um mesmo filme pode ser apresentado em diferentes salas e em horários diferentes. Cada apresentação em uma determinada sala e horário é chamada sessão. Um filme sendo apresentado em uma sessão tem um conjunto máximo de ingressos, determinado pela capacidade da sala.
- Os clientes do cinema podem comprar ou não ingressos para assistir a uma sessão. O funcionário deve intermediar a compra do ingresso. Um ingresso deve conter informação como o tipo de ingresso (Meio ingresso ou ingresso inteiro). Além disso, um cliente só pode comprar ingressos para sessões ainda não encerradas.

## 2.3 Sistema de Controle de Hotelaria

Requisitos gerais:

- Os quartos podem ser alugados no momento em o hóspede chega ao hotel (desde que existam vagas) ou serem reservados via internet.
- Caso seja a primeira vez que aluga quartos, ou seus dados tenham mudado, o hóspede deve ser cadastrado antes de finalizar o aluguel do quarto.
- Além do aluguel do quarto, o hotel oferece diversos serviços, como restaurante, lavar e/ou passar roupa etc. Obviamente, qualquer desses serviços se solicitados, será cobrado na fatura final.
- O hóspede pode também consumir os produtos contidos no frigobar, que também são cobrados pelo hotel.
- As diárias vencem ao médio dia.
- A política do hotel exige que as diárias sejam quitadas semanalmente. Quando o cliente for quitar a fatura, quitará não somente as diárias do(s) quartos que alugo, mas também qualquer serviço que tenha solicitado e os itens consumidos no frigobar.
- O hóspede, depois de quitar a fatura, pode permanecer no hotel ou encerrar sua estadia.
- Quando for encerrar sua estadia, o hóspede deverá pagar quaisquer serviços ou diárias ainda não pagas.

## 2.4 Sistema de Locadora de Jogos de Vídeo Game

Cada jogo e consola possui seu preço diário de locação, sendo que um mesmo jogo pode ter preços de locação diferentes para cada plataforma (Xbox, PS3, PS4, PC, etc.). O cliente (nome, identidade, cpf e-mail, telefone) especifica o jogo, plataforma e dias (pode alocar vários jogos de diferentes plataformas por vários dias). A data e hora da locação são armazenadas.

## 3 Projeto UML

A partir desse momento começamos a discutir e detalhar os projetos UML descritos acima

### **3.1 UML: Locadora de Carro**

#### **3.1.1 Diagrama de Caso de Uso**

#### **3.1.2 Diagrama de Classes**

#### **3.1.3 Diagrama de Instâncias**

#### **3.1.4 Diagrama de Sequências**

### **3.2 UML: Sistema de Cinema**

#### **3.2.1 Diagrama de Caso de Uso**

#### **3.2.2 Diagrama de Classes**

#### **3.2.3 Diagrama de Instâncias**

#### **3.2.4 Diagrama de Sequências**

### **3.3 UML: Sistema de Controle de Hotelaria**

#### **3.3.1 Diagrama de Caso de Uso**

#### **3.3.2 Diagrama de Classes**

#### **3.3.3 Diagrama de Instâncias**

#### **3.3.4 Diagrama de Sequências**

### **3.4 UML: Sistema de Locadora de Jogos de Vídeo Game**

#### **3.4.1 Diagrama de Caso de Uso**

#### **Aluguel de um Jogo**

##### **Caso de uso Aluguel de um Jogo**

Cenário principal de sucesso

1. Cliente navega pelo catalogo e seleciona o item para alugar
2. Segue até a o caixa
3. Cria conta do Cliente
4. O cadastro do cliente é aceito
5. Sistema apresenta informação completo do faturamento do empréstimo, incluindo preço e dias de empréstimo.
6. Cliente realiza o pagamento.
7. Sistema registra o pagamento e autoriza o empréstimo ao cliente, é gerado recibo de empréstimo para o cliente.
8. Cliente leva o produto por X dias
9. Após o período de locação o cliente retorna com o jogo para devolução
10. Sistema verifica que está no prazo
11. Sistema verifica integridade dos jogos

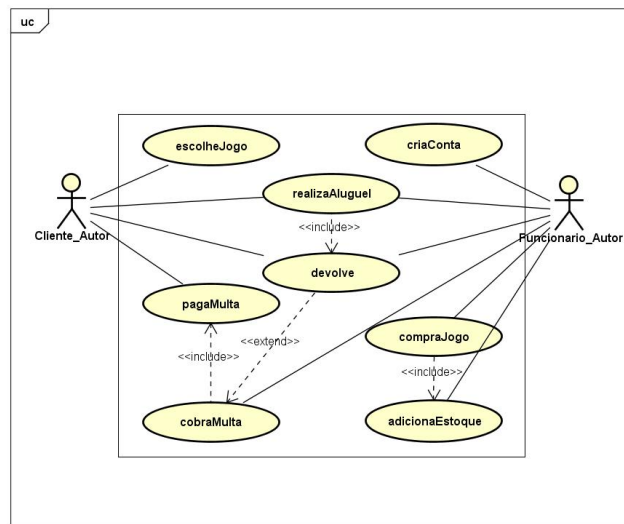


Figura 1: diagrama de uso para o sistema de controle de Jogos

12. Sistema finaliza o empréstimo e emite recibo de devolução para o cliente.

#### Extensões

(3) Se o cliente é regular

1. Cliente já possui cadastro
2. Sistema procura por débitos anteriores.
  - (a) Se identificado débitos anteriores, exige quitação do debito
  - (b) Senão autoriza a nova Locação.
3. pula para item 5.
- (10) Sistema identifica que o empréstimo passou do período de locação
  1. Sistema cobra multa do Cliente.
  - (11) Sistema verifica que os jogos estão danificados
    1. Sistema cobra Multa do Cliente por danos ao patrimônio.

Veja na figura 1 o diagrama de caso de uso para esse problema

#### 3.4.2 Diagrama de Classes

#### 3.4.3 Diagrama de Instâncias

#### 3.4.4 Diagrama de Sequências

## Referências

- [1] Jose Henrique Lincoln da Fonseca Felizardo. Historia da uml. <https://www.projetodiario.net.br/historia-da-uml>, julho 2013. [Online; acessado em 20 de marco de 2019].

ITEM	VALUE
UseCase	adicionaEstoque
Summary	Adiciona o jogo comprado ao estoque da loja
Actor	Funcionario_Autor
Precondition	
Postcondition	
Base Sequence	
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

(a) Adiciona Estoque

ITEM	VALUE
UseCase	cobraMulta
Summary	Cobra multa do Cliente por atraso ou danificar Jogo
Actor	Funcionario_Autor
Precondition	Atraso na entrega do jogo ou entregar jogo danificado
Postcondition	
Base Sequence	
Branch Sequence	
Exception Sequence	
Sub UseCase	pagaMulta
Note	

(b) Cobra Multa

ITEM	VALUE
UseCase	compraJogo
Summary	Realiza a compra de um jogo novo para a loja
Actor	Funcionario_Autor
Precondition	O jogo precisa já ter sido lançado
Postcondition	Adiciona o jogo no estoque da loja e disponibiliza para aluguel
Base Sequence	
Branch Sequence	
Exception Sequence	
Sub UseCase	adicionaEstoque
Note	

(c) compra Multa

ITEM	VALUE
UseCase	criaConta
Summary	Abre uma conta de um Cliente Novo
Actor	Funcionario_Autor
Precondition	Verifica se a conta do Cliente já existe
Postcondition	
Base Sequence	
Branch Sequence	
Exception Sequence	Na falta de Dados cadastrais não finaliza o cadastro
Sub UseCase	
Note	

(d) Cria Jogo

Figura 2: Casos de uso Sistema de locadora de Jogos

ITEM	VALUE
UseCase	devolve
Summary	O cliente devolve o jogo para a loja
Actor	Cliente_Autor, Funcionario_Autor
Precondition	O estado de conservação do jogo precisa estar igual ao sair da loja
Postcondition	
Base Sequence	
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

(a) Devolve Jogo

ITEM	VALUE
UseCase	escolheJogo
Summary	Escolhe o jogo
Actor	Cliente_Autor
Precondition	O jogo precisa estar disponível na loja
Postcondition	
Base Sequence	
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

(b) Escolhe Jogo

ITEM	VALUE
UseCase	pagaMulta
Summary	cliente paga a multa
Actor	Cliente_Autor
Precondition	ter sido cobrado
Postcondition	o débito é retirado do cliente
Base Sequence	
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

(c) Paga Multa

ITEM	VALUE
UseCase	realizaAluguel
Summary	Realiza o aluguel do jogo
Actor	Cliente_Autor, Funcionario_Autor
Precondition	O jogo precisa estar disponível na loja e o cliente precisa ter uma conta válida e sem débitos
Postcondition	o jogo é tornado indisponível na loja
Base Sequence	
Branch Sequence	
Exception Sequence	Conta Invalida de cliente, conta com pendencia de cliente, jogo indisponível
Sub UseCase	devolve
Note	

(d) Realiza Aluguel

Figura 3: Casos de uso Sistema de locadora de Jogos

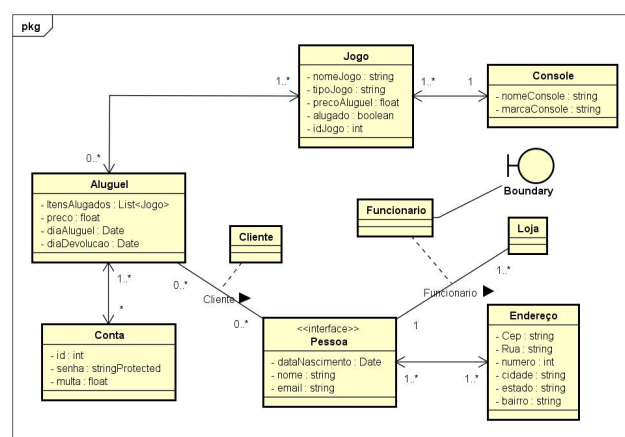


Figura 4: Diagrama de classe para o sistema de controle de jogos