



Disciplina	Curso	Semestre
Laboratório de Programação Modular	Ciência da Computação	2024/1º
Professor		
Hugo de Paula (hugo@pucminas.br)		

2024																																		
Fevereiro							Março							Abril							Maio							Junho						
S	T	Q	Q	S	S	D	S	T	Q	Q	S	S	D	S	T	Q	Q	S	S	D	S	T	Q	Q	S	S	D	S	T	Q	Q	S	S	D
			01	02	03	04					01	02	03	01	02	03	04	05	06	07				01	02	03	04	05					01	02
05	06	07	08	09	10	11	04	05	06	07	08	09	10	08	09	10	11	12	13	14	06	07	08	09	10	11	12	03	04	05	06	07	08	09
12	13	14	15	16	17	18	11	12	13	14	15	16	17	15	16	17	18	19	20	21	13	14	15	16	17	18	19	10	11	12	13	14	15	16
19	20	21	22	23	24	25	18	19	20	21	22	23	24	22	23	24	25	26	27	28	20	21	22	23	24	25	26	17	18	19	20	21	22	23
26	27	28	29				25	26	27	28	29	30	31	29	30						27	28	29	30	31			24	25	26	27	28	29	30

1 Especificação

1.1 Sprint 1

Objetivos:

modelo UML (agregação), classes de negócio e testes unitários, versão 1 do sistema.

Um restaurante atende seus clientes alocando-os em mesas por meio de 'requisições por mesas'.

No momento, o restaurante possui 10 mesas: 4 com capacidade para 4 pessoas, 4 com capacidade para 6 pessoas e 2 com capacidade para 8 pessoas. O restaurante deve ser capaz de ampliar a sua capacidade com o menor esforço possível.

O cliente, ao chegar, deve declarar quantas pessoas comerão no restaurante, gerando assim uma requisição por mesa.

Tão logo quanto possível deverá ser alocada uma mesa adequada para ele. Se não houver mesa livre, o cliente entra numa fila de espera. As requisições precisam registrar as datas e horas de entrada e saída do cliente.

1.2 Sprint 2

Objetivos:

atualizar projeto UML, novas classes e testes, novas regras de negócio e persistência/spring.

O restaurante oferece aos clientes um cardápio simplificado com 6 pratos e 5 opções de bebidas como por exemplo:

- Moqueca de Palmito
- Falafel Assado
- Salada Primavera com Macarrão Konjac
- Escondidinho de Inhame
- Strogonoff de Cogumelos
- Caçarola de carne com legumes

- Água
- Suco
- Refrigerante
- Cerveja
- Taça de vinho

Cada mesa, enquanto está sendo servida, pode fazer um pedido com quantos pratos e bebidas desejar. Novos produtos podem ser adicionados ao pedido no decorrer do atendimento.

Ao encerrar o pedido, a conta incluirá uma taxa de serviço de 10% e exibirá tanto o valor total como o valor a ser dividido igualmente entre os ocupantes.



1.3 Sprint 3

Objetivos:

atualizar projeto UML, novas classes e testes com herança e/ou composição com interface.

Cada conta pode ser paga por diferentes métodos: dinheiro, Pix, débito ou crédito. O que muda para cada método são o prazo para receber o valor e a taxa que o restaurante paga ao banco por este recebimento. Além disso, para pagamento em Pix, deve ser armazenado o nome do emitente do Pix. No caso de débito, deve ser armazenado o nome do banco de origem do débito. Finalmente, no caso de cartão de crédito, deve ser armazenado a bandeira do cartão (Visa, Mastercard, Elo, American Express).

- **dinheiro:** prazo 0 dias, desconto 0%
- **pix:** prazo 0 dias, desconto de 1,45%, limitado a R\$10
- **débito:** prazo 14 dias, desconto de 1,4%
- **crédito:** prazo 30 dias, desconto de 3,1%

O restaurante precisa saber quanto vendeu em um dia e quanto receberá em datas futuras a serem determinadas pela gerência.

Pedidos para delivery não incluem taxa de serviço, mas incluem taxa de entrega de acordo com a distância: Estes pedidos não precisam especificar o valor individual a ser pago em uma conta.

1.4 Sprint 4

Objetivos:

atualizar projeto UML, novos requisitos para incorporar tratamentos de erros, exceções e padrões de projeto.

Esse trabalho deve usar padrões de projeto, como por exemplo: **Singleton**, **Decorator**, **Factory** e **Observer**.

1.5 Critérios de correção

Será avaliado:

- a conformidade com os requisitos
- qualidade do código: correção e robustez
- modularização: alta coesão e baixo acoplamento, extensibilidade e adequação aos princípios SOLID
- utilização dos conceitos básicos e avançados de programação modular: tais como polimorfismo, collections, streams e tratamento de exceções.
- testes unitários baseados na JUnit.



2 Cronograma de realização de tarefas

SEXTA-FEIRA	
Março 19 Sprint 1 – Diagrama de Classes na UML	1
26 Semana Santa – Recesso	
Abril 2 Sprint 1 – Implementação classes de negócio e testes unitários, versão 1 do sistema. Apresentação da Sprint 1	2
9 Aula de SpringBoot Sprint 2 – implementação dos novos requisitos.	3
16 Aula de Herança e polimorfismo Sprint 2 – implementação da persistência / spring.	4
23 Aula de Interfaces Sprint 2 – implementação dos testes.	5
30 Apresentação da Sprint 2	6
Maió 7 Aula de Collections e Streams. Sprint 3 – implementação de polimorfismo e interfaces.	7
14 Sprint 3 – implementação de collections e streams.	8
21 Aula de Padrões de Projeto	9
28 Apresentação da Sprint 3	10
Junho 4 Aula de Eventos e interface gráfica Sprint 4 – implementação dos padrões de projeto.	11
11 Sprint 3 – implementação dos tratamentos de erro.	12
18 Apresentação da Sprint 4	13



SEXTA-FEIRA

25

Reavaliação

14