



PROJETO INTEGRADOR DE SISTEMAS [7º]

por Rodrigo Costa Bittencourt Silva Arruda

rodrigo.arruda@aluno.cefet-rj.br

29 de janeiro de 2025

CONTEXTO

Foi nos incumbida a tarefa de desenvolvermos um sistema de reserva de horários para um restaurante.

Esse sistema precisaria ser criado por meio de uma aplicação web acessível tanto para *desktops* como para celulares.

Ethan Marcotte popularizou o termo “responsividade” em 2010, em seu artigo “Responsive Web Design”, como a prática de criar páginas web que respondam às necessidades dos usuários e ao ambiente do dispositivo no qual estão sendo visualizadas [1].

[1] MARCOTTE, Ethan. Responsive Web Design. A List Apart, 2010. Disponível em: <https://alistapart.com/article/responsive-web-design/>. Acesso em: 7 nov. 2024.

REQUISITOS

- Deverá ser possível reservar uma mesa em horários não conflitantes;
- Deverá ser possível listar as reservas feitas;
- Deverá ser possível cancelar uma reserva; e,
- Deverá ser possível gerar um relatório contendo a quantidade de reservas feitas por data.

Todos esses requisitos seriam desenvolvidos para o **funcionário** do restaurante, o qual, em horário de trabalho, realizaria as funcionalidades levantadas.

Barry W. Boehm popularizou o termo “requisitos funcionais de software” em 1981, em sua publicação na IEEE “Software Engineering Economics”, como especificações que detalham as funcionalidades que o sistema deve executar [2].

REQUISITOS (2)

- Deverá ser possível o funcionário realizar Login/Logout no sistema
 - Deverá ser possível lançar consumos de uma mesa.
 - Deverá ser possível o gerente fechar a conta de uma mesa.
 - Deverá ser possível o gerente gerar outros quatro relatórios: vendas por período e forma de pagamento; vendas por período e funcionário; vendas por período e categoria; e, vendas por dia.
-

REQUISITOS (3)

- Movimentação de Estoque (entrada/saída).
 - Listagem de Movimentações de Estoque.
 - Visualização de Movimentação de Estoque.
-

INTERFACE

O Responsive Web Design (RWB) adota três componentes principais:

- Layouts Flexíveis: Utilizam grades fluidas, ou seja, em vez de usar unidades fixas como pixels, utilizam unidades relativas como porcentagens para definir as dimensões dos elementos.
- Imagens Flexíveis: As imagens e outros conteúdos multimídia devem ser escaláveis e ajustar-se proporcionalmente ao tamanho da tela.
- Media Queries: Utilizam-se para aplicar diferentes estilos a diferentes tamanhos de tela, resoluções, e outras características do dispositivo.

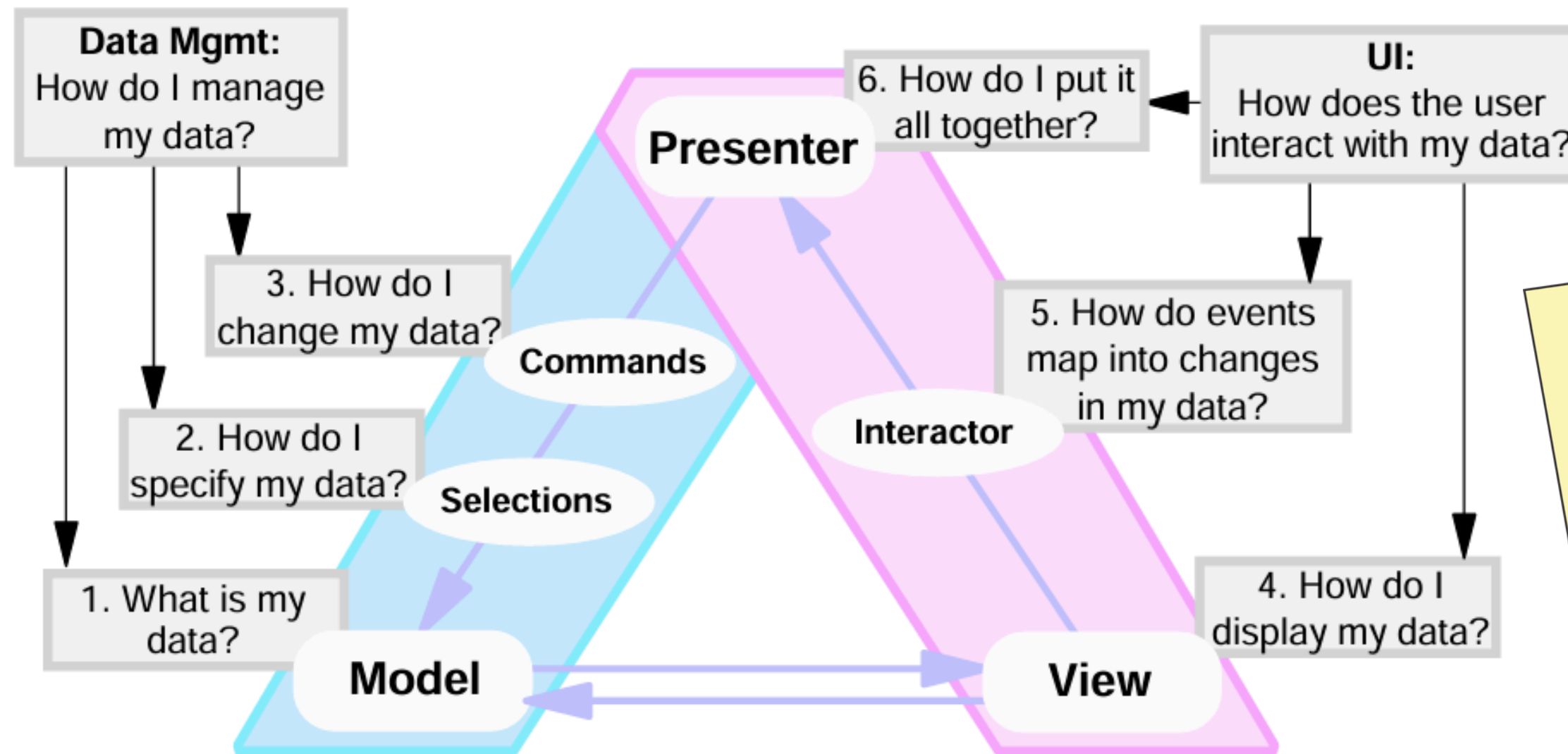
Como uma biblioteca de estilos que adota em suas funcionalidades tais princípios, decidimos adotar o **Bootstrap 5** [3] para o desenvolvimento da interface da aplicação.

Bootstrap é uma estrutura organizada que fornece um conjunto de diretrizes, convenções e padrões para desenvolvimento de interfaces de software.

[3] BOOTSTRAP. Powerful, extensible, and feature-packed frontend toolkit1. Disponível em: <https://getbootstrap.com>. Acesso em: 7 nov. 2024.

ARQUITETURA

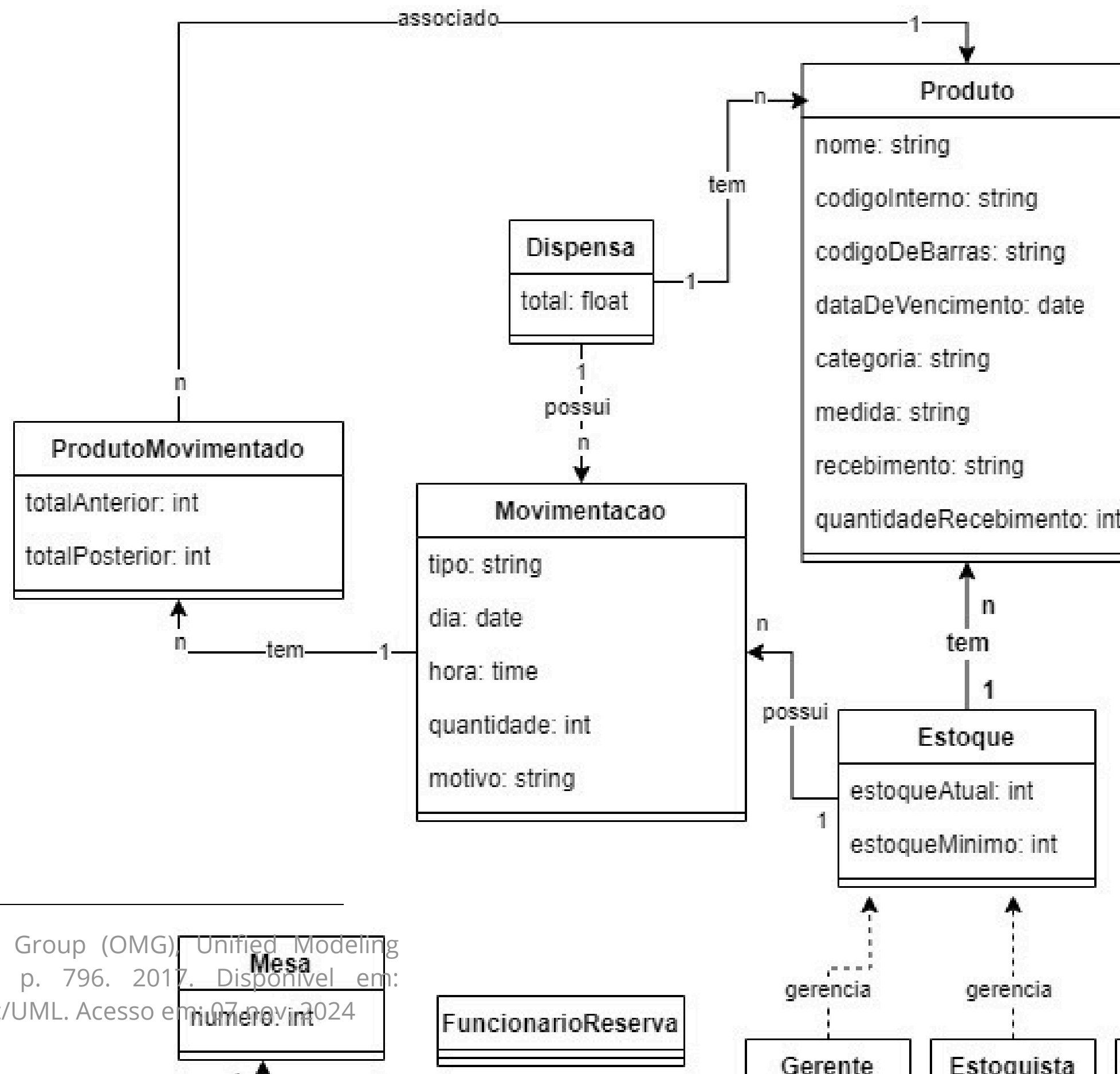
Para a arquitetura do sistema foi utilizado o padrão MVP:



O MVP (Model-View-Presenter) é essencialmente uma generalização do padrão MVC (Model-View-Controller) para representar uma estrutura geral de qualquer programa interativo [4].

[4] Potel, Mike. "MVP: Model-View-Presenter the Taligent programming model for C++ and Java." Taligent Inc 20 (1996). Disponível em: <https://www.wildcrest.com/Potel/Portfolio/mvp.pdf>. Acesso em: 7 nov. 2024.

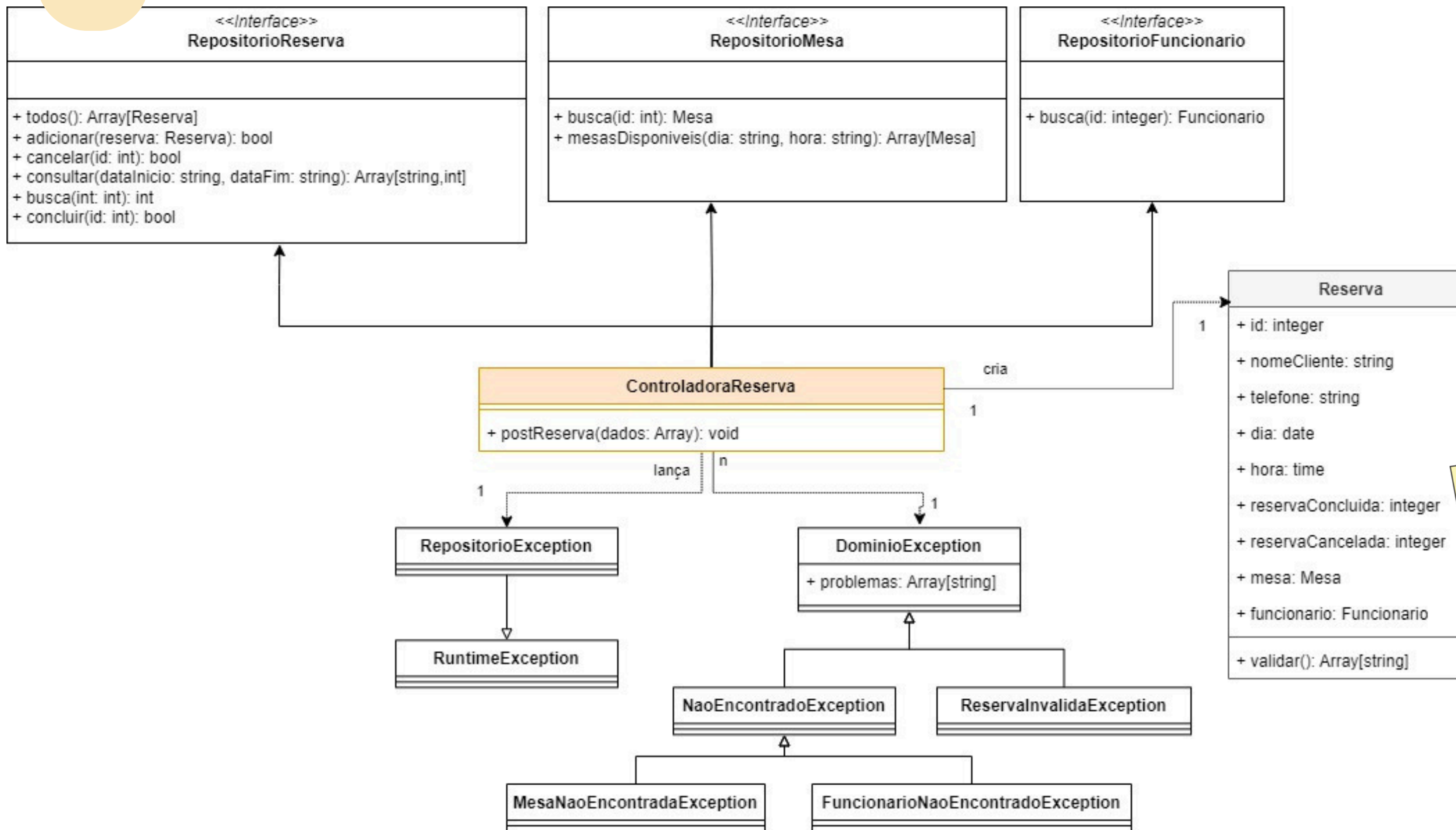
DIAGRAMA DE CLASSES UML ESSENCIAL



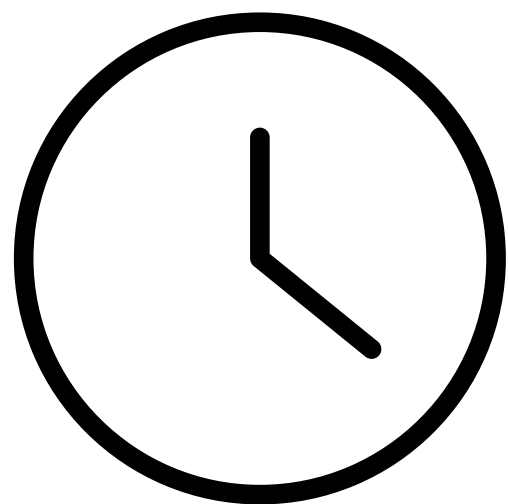
O diagrama UML (Unified Modeling Language) [5] foi desenvolvido na década de 1990 por três especialistas em engenharia de software: Grady Booch, James Rumbaugh e Ivar Jacobson. Eles uniram suas metodologias de modelagem (Booch Method, OMT e OOSE) para criar uma linguagem de modelagem unificada que pudesse ser usada para representar sistemas de software de maneira padronizada e compreensível.

[5] Object Management Group (OMG), Unified Modeling Language (OMG UML), p. 796. 2017. Disponível em: <https://www.omg.org/spec/UML>. Acesso em: 07 nov. 2024

DIAGRAMA DE CLASSES UML PARCIAL



A funcionalidade principal do sistema era a de **cadastro de uma reserva**. Portanto foi realizado este recorte do ponto de vista da **ControllerReserva** - responsável pela execução do **postReserva** - para o Diagrama Parcial - que se preocupa com a arquitetura de implementação e de detalhes "de mais baixo nível" da funcionalidade principal.



**(ESPAÇO PARA
EXPOSIÇÃO)**