U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Bases de Dados

Gestão de um Hotel

Grupo 1003:

João Maria Correia Rebelo up202107209@fe.up.pt

Carlos Filipe Oliveira Sanches Pinto up202107694@fe.up.pt

Pedro Afonso Nunes Fernandes up202207987@fe.up.pt

Contexto da Base de Dados:

- Este projeto tem como objetivo principal o desenvolvimento de uma solução eficiente para otimizar a administração de um Hotel, o modelo UML aplicado facilita a visualização das entidades envolvidas neste sistema bem como a sua estrutura e relações. Não sendo um modelo estático, garante a fácil adaptação deste hotel a uma constante evolução do setor hoteleiro, e permite também uma certa facilidade em escalar este projeto.

- Comecemos por considerar a classe **Pessoa**, que é responsável por armazenar informações detalhadas sobre cada indivíduo relacionado ao hotel, atributos como o NIC (número de identificação civil) permitem manter a unicidade de cada indivíduo garantindo a integridade dos dados que lhe estão associados. Serve também como suporte a uma generalização sendo superclasse para as subclasses “Funcionário” e “Hóspede”.

- Seguem-se as classes **Funcionário** e **Hóspede**, que por inerência herdam os atributos da classe anterior, evitando assim a repetição de atributos que lhes seriam comuns. Esta generalização é completa e disjunta, pois todas as pessoas representadas nesta base de dados ou são hóspedes ou funcionários.

- Para cada Funcionário fica registada uma data de início e fim de contrato explícitas no seu **Contrato**, que pode ou não ser a tempo inteiro, este fator tem impacto naquilo que vai ser o seu salário. As restrições associadas a esta classe estabelecem o salário mínimo legal que pode ser atribuído a um funcionário se este trabalhar a tempo inteiro, tal como a fórmula utilizada para calcular o salário de um funcionário com um contrato em part-time.

- Este Funcionário pode ser alocado a **Restaurante Staff** ou a **Hotel Staff**, sendo estas subclasses da superclasse Funcionário. Para os funcionários alocados à staff do hotel é importante a base de dados registar a lista de idiomas falados, tal como o seu local de trabalho no Hotel. Sobre os trabalhadores da área do restaurante é relevante saber qual o seu cargo de trabalho.

- O Hotel contém também um serviço independente de **Restaurante**, com um horário fixo, e um menu variável e inclusivo registado na nossa base de dados.

- Associamos a classe Hóspede à classe **Reserva** visto que cada Hóspede tem a possibilidade de efetuar a sua reserva.

- Ligamos a Reserva a **Estadia**, mas é de ressaltar que a reserva pode ser cancelada não acontecendo qualquer estadia por parte do hóspede. É importante guardar um idReserva que identifica cada uma das reservas e na classe Estadia uma Data para o check-in e o check-out tal como o atributo derivado, duração que resulta da diferença entre estas duas datas.

- No fim de cada Estadia o Hóspede pode, se assim desejar, efetuar uma **Avaliação** atribuindo um valor que varia entre 0 e 5 que reflete a sua experiência com o serviço prestado, assim como um comentário.

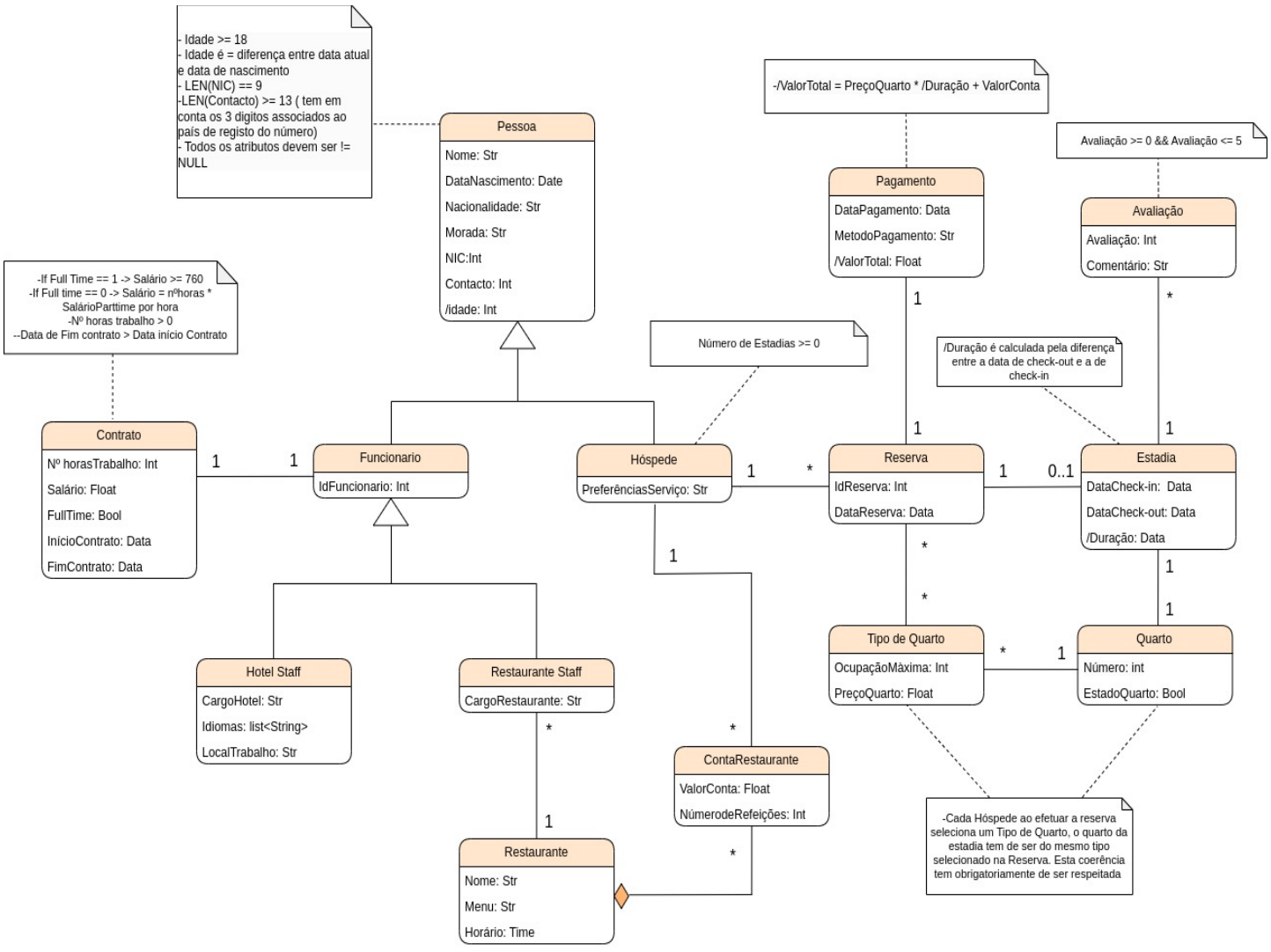
- Ao efetuar a Reserva o hóspede deve selecionar o **Tipo de Quarto** que deseja, sendo-lhe depois, no momento da Estadia alocado um quarto específico que deve ser sempre do mesmo tipo selecionado na reserva (restrição explícita no UML). A classe **Quarto** armazena o

número do Quarto, e mantém atualizado o estado deste através de um booleano que indica se está ocupado (1) ou não (0).

- O Quarto pode ter vários Tipos de Quarto que definem a sua ocupação máxima e o seu preço por noite.

- Resta a classe **Pagamento** que tem como objetivo manter atualizado o valor total a pagar pelo Hóspede, valor este que deriva da duração da estadia a multiplicar pelo preço do tipo de quarto por noite, tal como manter registado o método de pagamento utilizado, e a data deste pagamento.

Diagrama UML:



Utilização de AI:

Neste ponto do trabalho recorremos a um modelo de linguagem natural avançado o GPT 4.0, de modo a tentar melhorar a nossa proposta inicial de diagrama, e avaliar as potencialidades desta ferramenta.

Início Input

Tenho o seguinte diagrama uml :

Pessoa: (superclass) (atributos) Nome:Str; Data de Nascimento: Date; Nacionalidade: String; Morada: string ; NIC: int ; Contacto: string ; /idade: int

Funcionário :(subclasse de pessoa) (superclass) (atributos) IdFuncionario:int;

Contrato: (atributos) Full time: bool ; Dias de Trabalho: list <str>; Salário: float; Data início contrato: Data; Data fim contrato: Data

Hóspede: (atributos) PreferênciasServiço: String

Hotel Staff :(subclasse de funcionário) (atributos)CargoHotel: String; Idiomas: String; Local Trabalho: String

Restaurante Staff :(subclasse de funcionário) (atributos)CargoRestaurante:String

Restaurante: (atributos) Nome: string; Menu:String; Horário: Time

Conta Restaurante: (composição com o restaurante / depende da existência de restaurante) (atributos)Valor: float ; Número refeições: int

Reserva: (atributos) IDreserva : int; DataReserva : Data

Estadia: (classe que que por aggregation associa-se a reserva) (atributos) Data check-in: Data; Data check-out: Data

Quarto: (atributos) Número: int ; Estado do quarto: bool

Tipo de Quarto: (atributos) Ocupação Máxima: int ; Preco quarto: float

Avaliação: (atributos) Avaliação: int ; Comentário: str

Pagamento: (atributos) DataPagamento : Data; MetodoPagamento : str; /valor total: float

Seguem-se as associações:

Contrato 1 <-> 1 Funcionário

Restaurante Staff * <-> 1 Restaurante

Restaurante 1 <-> * ContaRestaurante

Quarto 1 <-> 1 Tipo de quarto

Reserva 1 <-> 0..1 Estadia

TIpo de Quarto * <-> * Reserva

Estadia 1 <-> * Avaliação

Estadia 1 <-> 1 Quarto

Hóspede 1 <-> * Conta Restaurante

Hóspede 1< -> * Reserva

Pagamento 1 <-> 1 Reserva

TIpo de Quarto * <-> 1 Quarto

Passa o diagrama que forneci para a linguagem plantUML e faz melhorias ao meu modelo.

FIM INPUT

-Output do GPT 4.0

Nesta versão, adicionei cinco novas classes: Serviço: Esta classe representa os serviços adicionais que o hotel pode oferecer aos hóspedes, como limpeza, serviço de quarto, spa, etc. Cada hóspede pode ter zero ou mais serviços associados.

Evento: Esta classe representa eventos que o hotel pode hospedar, como conferências, casamentos, etc. Cada hóspede pode participar de zero ou mais eventos.

Gerente: Esta classe representa gerentes que trabalham na organização. Eles podem administrar diferentes departamentos, o que é representado pelo atributo departamento.

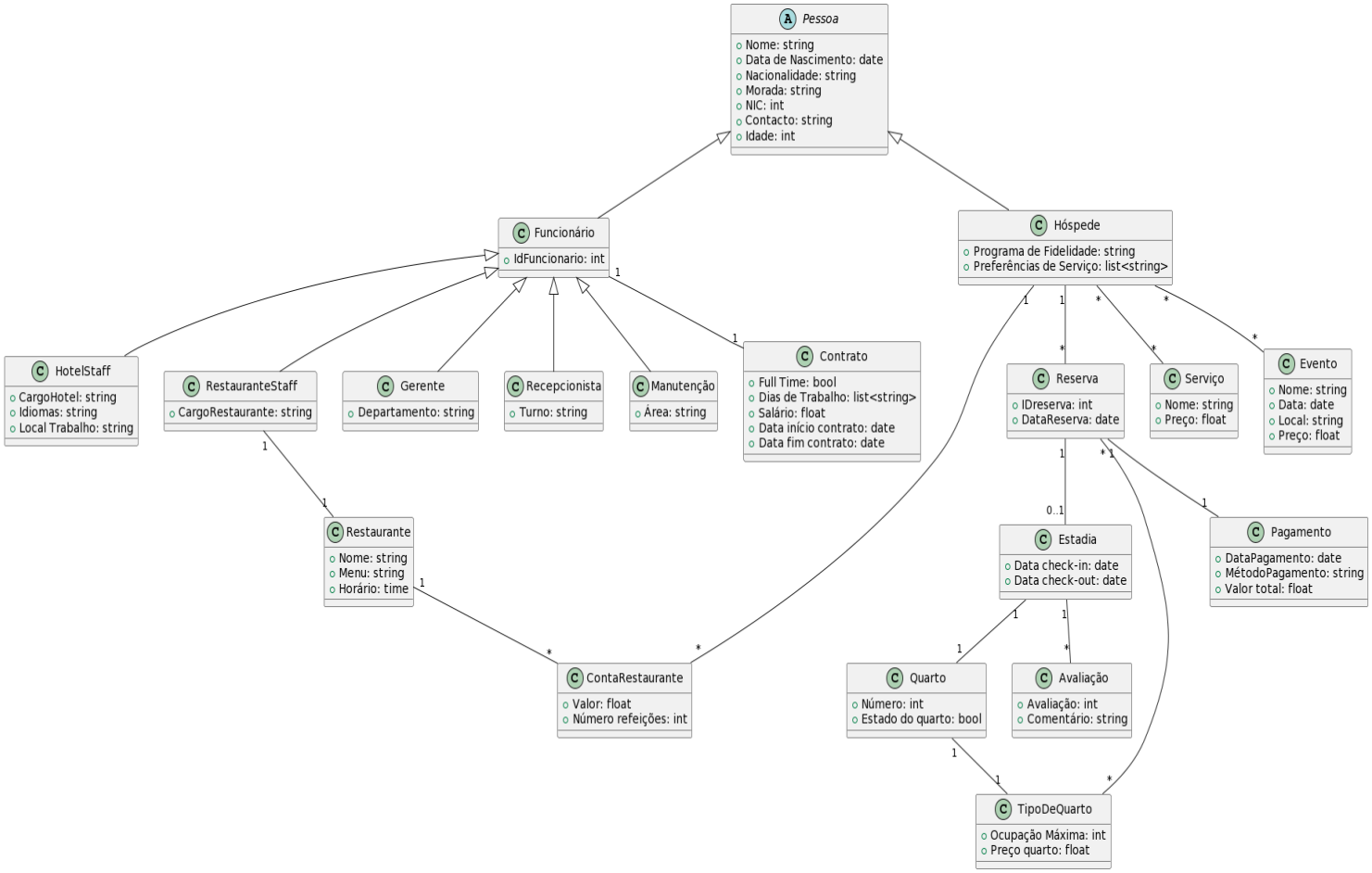
Rececionista: Esta classe representa os rececionistas da organização. Eles podem trabalhar em diferentes turnos, que é representado pelo atributo turno.

Manutenção: Esta classe representa os funcionários de manutenção. Eles podem ter uma área de especialização, que é representada pelo atributo areaEspecializacao.

Além disso, adicionei associações entre Hóspede e as novas classes Serviço e Evento para representar o fato de que os hóspedes podem utilizar serviços e participarClaro, aqui está uma versão expandida do seu diagrama UML, onde adicionei algumas classes novas para representar conceitos adicionais que poderiam ser relevantes para um sistema de gerenciamento de hotel:

(O chat GPT forneceu também um extenso código na linguagem PlantUML que pode ser consultado em anexo à entrega do projeto, e que permitiu a visualização das melhorias feitas pela ferramenta)

Proposta Final de UML:



Refinamentos feitos pela IA e crítica:

- A ferramenta de IA utilizada permitiu que o modelo apresentasse uma maior diversidade de serviços a prestar no Hotel, tal como um maior detalhe acerca do trabalho desempenhado pelos funcionários.

- Ao desempenhar este trabalho de refinamento conseguiu manter muitas das características do modelo inicialmente proposto, e entender também a lógica de funcionamento de toda a gestão do hotel.

- Uma das funcionalidades mais interessantes, foi a possibilidade de pedir ao GPT 4 para passar a sua proposta de modelo final para a linguagem PlantUML, uma linguagem open source que permite a visualização de diagramas UML. Colocamos o código de output fornecido pelo modelo de IA no site https://plantuml.com/ que gerou o diagrama da segunda imagem, é de notar a facilidade com que é possível visualizar as mudanças feitas, torna o processo de visualização mais rápido do que a construção que foi feita inicialmente através do draw.io.

- Ainda que esta tecnologia apresente inúmeras vantagens no que toca ao aumento de produtividade do trabalho, com a sua melhoria do modelo em poucos segundos e também a facilidade de visualização de todas as melhorias feitas, não deixa de ser uma ferramenta limitada pois toda esta sua capacidade de abstração tem como base um modelo com regras que definimos ao compor o modelo inicial.

-Concluímos assim que as capacidades de integração da IA servem um propósito não de criação na integra de soluções para problemas, mas sim um apoio de carater incremental na resolução destes.