

Faculdade de Engenharia da Universidade do Porto



# Gestão de Alojamentos Locais

Projeto da Unidade Curricular de Bases de Dados

**Turma:** 2LEIC06

**Grupo:** 602

**Trabalho realizado por:**

Anete Pereira (up202008856)

Isabel Amaral (up202006677)

Tomás Pires (up202008319)

# Índice

<b>1. Contextualização</b>	<b>3</b>
<b>2. Modelo Conceptual</b>	<b>4</b>
<b>3. Modelo Relacional</b>	<b>5</b>
<b>4. Dependências Funcionais</b>	<b>6</b>
<b>5. Análise das formas normais</b>	<b>7</b>
<b>6. Restrições</b>	<b>9</b>
<b>7. Interrogações À Base de Dados</b>	<b>12</b>
<b>8. Gatilhos</b>	<b>13</b>
<b>9. Avaliação da participação de cada elemento do grupo</b>	<b>14</b>

# 1. Contextualização

Nos últimos anos, os alojamentos locais têm ganho popularidade em virtude dos seus preços mais acessíveis em relação aos hotéis convencionais. Como tal, são uma opção atrativa para muitos, tanto como estadia para férias ou durante viagens em trabalho. Esta alta procura levou ao sucesso de várias plataformas online direcionadas para a reserva de alojamentos locais, nomeadamente o Airbnb.

Neste trabalho pretende-se a criação de uma base de dados que armazene informação relativa aos alojamentos locais na cidade de Nova Iorque.

Existem tipos de alojamentos para todos os gostos. Um alojamento pode ser um apartamento ou, se falarmos de opções mais baratas, um quarto privado ou até partilhado.

Em plataformas como o Airbnb, cada alojamento tem uma descrição geral da propriedade e uma descrição das várias características da casa, tais como número de quartos, camas e casas de banho, e das suas comodidades (se tem wifi, televisão, aquecedor, detetor de fumo...), de forma a permitir que os possíveis clientes reservem algo de acordo com as suas necessidades.

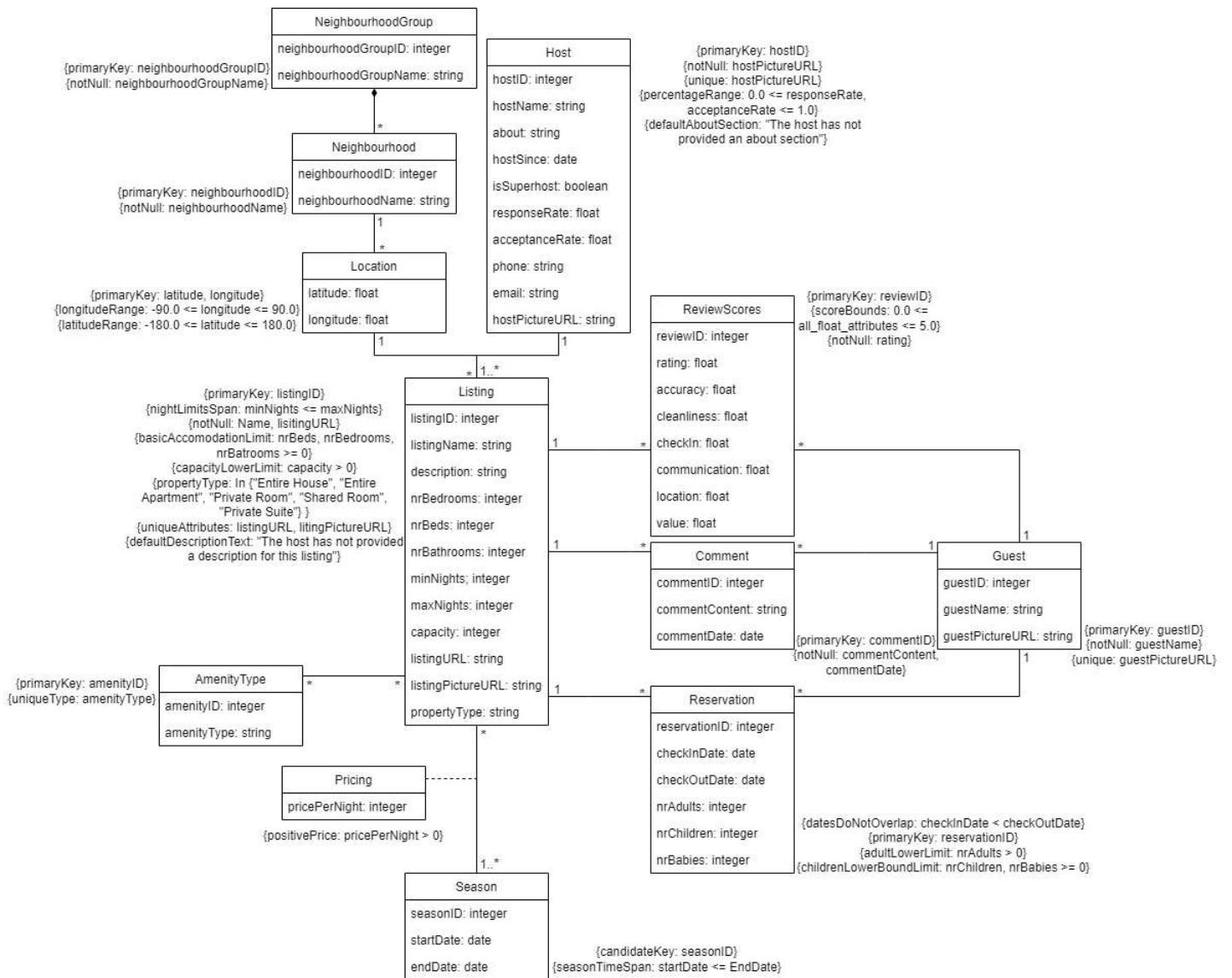
O responsável por um dado alojamento e acolhimento dos seus hóspedes é o anfitrião. Do anfitrião, os hóspedes terão que ter acesso a informações importantes tais como a data que começou a hospedar, o tempo médio de resposta e se é ou não um superhost. Uma breve descrição sobre o mesmo pode em alguns casos também ser útil para uma maior familiarização entre os hóspedes e o respetivo anfitrião.

A disponibilidade do alojamento vai depender do número de reservas e respetivas datas de check-in e check-out. Cada reserva deve cumprir um conjunto de regras definidas pelo anfitrião, isto é, deve respeitar o número mínimo e máximo de noites permitido, bem como a capacidade máxima. É de acordo com o tempo de estadia escolhido pelo hóspede, que se calcula o total a pagar de acordo com o preço por noite, que depende da época corrente.

Após a estadia, os hóspedes são recomendados a deixar um comentário sobre o alojamento onde estiveram hospedados. Este comentário é importante tanto para promover o alojamento, como para confirmar a veracidade das informações disponíveis na plataforma sobre o mesmo.

Além do comentário, é essencial que os hóspedes deixem a sua classificação em relação ao estabelecimento. Desta forma, cada alojamento possui uma série de classificações deixadas por cada hóspede de 0 a 5: uma classificação geral e várias outras opcionais, focadas em parâmetros específicos, tais como a precisão, limpeza, check-in, comunicação, localização e qualidade-preço.

## 2. Modelo Conceptual



### 3. Modelo Relacional

Listing (listingID, listingName, description, listingURL, listingPictureURL, nrBedrooms, nrBeds, nrBathrooms, minNights, maxNights, capacity, propertyType, hostID->Host, latitude->Location.latitude, longitude->Location.longitude)

Host (hostID, hostName, about, hostSince, isSuperhost, responseRate, acceptanceRate, phone, email, hostPictureURL)

Guest (guestID, guestName, guestPictureURL)

Comment (commentID, commentDate, commentContent, listingID->Listing, guestID->Guest)

ReviewScores (reviewID, rating, accuracy, cleanliness, checkIn, communication, location, value, guestID->Guest, listingID->Listing)

Reservation (reservationID, checkInDate, checkOutDate, nrAdults, nrChildren, nrBabies, listingID->Listing, guestID->Guest)

Location (latitude, longitude, neighbourhoodID->Neighbourhood)

Neighbourhood (neighbourhoodID, neighbourhoodName, neighbourhoodGroupID->NeighbourhoodGroup)

NeighbourhoodGroup (neighbourhoodGroupID, neighbourhoodGroupName)

Season (seasonID, startDate, endDate)

Pricing (listingID->Listing, SeasonID->Season, pricePerNight)

AmenityType (amenityID, amenityType)

hasAmenity (listingID->Listing, amenityID->AmenityType)

## 4. Dependências Funcionais

listingID->listingName, description, nrBedrooms, nrBeds, nrBathrooms, minNights, maxNights, capacity, listingURL, listingPictureURL, propertyType, hostID, latitude, longitude, neighbourhoodID, neighbourhoodGroupID

listingURL->listingID

listingPictureURL -> listingID

hostID -> hostName, hostSince, about, isSuperhost, responseRate, acceptanceRate, phone, email, hostPictureURL

hostPictureURL -> hostID

guestID -> guestName, guestPictureURL

guestPictureURL -> guestID

commentID -> commentDate, contents, guestID, listingID

reviewID -> rating, accuracy, cleanliness, checkIn, communication, location, value, guestID, listingID

seasonID -> startDate, endDate

listingID, seasonID -> pricePerNight

reservationID -> checkInDate, checkOutDate, nrAdults, nrChildren, nrBabies, listingID, guestID

latitude, longitude -> neighbourhoodID

neighbourhoodID -> neighbourhoodName, neighbourhoodGroupID

neighbourhoodGroupID -> neighbourhoodGroupName

amenityID -> amenityType

amenityType -> amenityID

## 5. Análise das formas normais

**Listing:** listingID constitui a chave primária da relação, logo trata-se de uma chave candidata.

Tanto listingURL como pictureURL são atributos únicos, estes determinam listingID, que por sua vez é uma chave da relação; como tal, pela regra da transitividade, determina-se que listingURL e pictureURL são chaves da relação.

Como todos os atributos do lado esquerdo das FDs se tratam de chaves da relação, conclui-se que a relação Listing obedece à Forma Normal de Boyce-Codd e à 3ª Forma Normal.

**Host:** hostID está marcado como a chave primária da relação, logo é uma chave candidata.

hostPictureURL, sendo um atributo único, determina a chave primária da relação, logo este atributo também constitui uma chave candidata.

Como todos os atributos do lado esquerdo das FDs se tratam de chaves da relação, conclui-se que a relação Host obedece à Forma Normal de Boyce-Codd e à 3ª Forma Normal.

**Guest:** guestID trata-se da chave primária de Guest, logo é uma chave candidata da relação.

guestPictureURL, sendo um atributo único, determina a chave primária da relação, logo este atributo também constitui uma chave candidata.

Como todos os atributos do lado esquerdo das FDs se tratam de chaves da relação, conclui-se que a relação Guest não viola a Forma Normal de Boyce-Codd e a 3ª Forma Normal.

**Comment:** A única dependência funcional desta relação é definida pela sua chave primária, {commentID}, logo a relação obedece à Forma Normal de Boyce-Codd e à 3ª Forma Normal.

**ReviewScores:** A única dependência funcional desta relação é definida pela sua chave primária, {reviewID}, logo a relação obedece à Forma Normal de Boyce-Codd e à 3ª Forma Normal.

**Season:** A única dependência funcional desta relação é definida pela sua chave primária, {seasonID}, logo a relação obedece à Forma Normal de Boyce-Codd e à 3ª Forma Normal.

**Pricing:** A única dependência funcional desta relação é definida pela sua chave primária, {listingID, seasonID}, logo a relação obedece à Forma Normal de Boyce-Codd e à 3ª Forma Normal.

**Reservation:** A única dependência funcional desta relação é definida pela sua chave primária, {reservationID}, logo a relação obedece à Forma Normal de Boyce-Codd e à 3ª Forma Normal.

**Location:** A única dependência funcional desta relação é definida pela sua chave primária, {latitude, longitude}, logo a relação obedece à Forma Normal de Boyce-Codd e à 3ª Forma Normal.

**Neighbourhood:** A única dependência funcional desta relação é definida pela sua chave primária, {neighbourhoodID}, logo a relação obedece à Forma Normal de Boyce-Codd e à 3ª Forma Normal.

**NeighbourhoodGroup:** A única dependência funcional desta relação é definida pela sua chave primária, {neighbourhoodGroupID}, logo a relação obedece à Forma Normal de Boyce-Codd e à 3ª Forma Normal.

**Amenity:** amenityID constitui a chave primária da relação, logo trata-se de uma chave candidata.

amenityType determina a chave primária da relação, logo pela regra da transitividade existe também uma chave candidata {amenityType}.

Como todos os atributos do lado esquerdo se tratam de chaves da relação, conclui-se que a relação Amenity obedece à Forma Normal de Boyce-Codd e à 3ª Forma Normal.

**hasAmenity:** Os únicos atributos desta relação constituem a sua chave primária, {listingID, amenityID}, pelo que a única dependência funcional desta relação é trivial e os seus atributos são primos. Logo, conclui-se que a relação hasAmenity obedece à Forma Normal de Boyce-Codd e à 3ª Forma Normal.



## 6. Restrições

**Listing:** Cada alojamento é identificado pelo seu ID, dado que não pode haver dois alojamentos com o mesmo ID, pelo que listingID apresenta a restrição PRIMARY KEY.

Adicionalmente, também não pode haver dois alojamentos com o mesmo URL do alojamento e URL da foto. Os atributos listingURL e listingPictureURL apresentam a restrição UNIQUE.

Cada alojamento tem de ter um nome. listingName é NOT NULL. Além disso, o número de quartos, camas e casas de banho, número mínimo e máximo de noites permitidas por estadia e a capacidade máxima do alojamento devem ter valores bem definidos, pelo que os atributos nrBedrooms, nrBeds, nrBathrooms, minNights, maxNights e capacity são todos NOT NULL.

Caso o anfitrião do alojamento não tenha decidido incluir uma descrição do mesmo, o atributo description deverá ter como valor DEFAULT a string "The host has not provided a description for this listing."

Os únicos tipos de alojamentos existentes são: casa completa, apartamento completo, quarto privado ou quarto partilhado. Assim, para o atributo propertyType deverá ser usada a restrição CHECK (propertyType in ("Entire House", "Entire Apartment", "Shared Room", "Private Room")). Este atributo deve estar bem definido, pelo que propertyType é NOT NULL.

O número de quartos, camas, casas de banho de um alojamento e o número máximo de pessoas que pode alojar (capacidade) não podem ser identificados por um inteiro negativo. Para cada um destes atributos está presente a restrição CHECK(atributo >= 0).

Em relação ao número mínimo e máximo de noites permitidas para cada estadia, a representação também não poderá ser um inteiro negativo, pelo que também está presente a restrição mencionada anteriormente. Além disso, o número mínimo de noites não pode ser superior ao número máximo. Ambos os atributos apresentam a restrição CHECK(minNights <= maxNights).

Cada alojamento tem um anfitrião identificado pelo seu ID que deve estar bem definido. hostID é, portanto, chave estrangeira da relação e NOT NULL. Neste caso temos a restrição, FOREIGN KEY (hostID) REFERENCES Host(hostID).

Cada alojamento tem uma localização identificada pela respetiva latitude e longitude, as quais devem estar bem definidas. Ambos estes atributos constituem uma chave estrangeira composta e são NOT NULL. Está presente a restrição FOREIGN KEY (latitude, longitude) REFERENCES Location(latitude, longitude).

**Host:** Cada anfitrião é identificado pelo seu ID, dado que não pode haver dois anfitriões com o mesmo ID, pelo que hostID apresenta a restrição PRIMARY KEY.

Adicionalmente, também não pode haver dois anfitriões com o mesmo URL da foto de perfil. O atributo hostPictureURL apresenta a restrição UNIQUE.

Cada anfitrião tem um nome. hostName é NOT NULL.

Caso o anfitrião não tenha decidido incluir uma descrição sobre si, o atributo about deverá ter como valor DEFAULT a string "The host has not provided an about section."

A informação acerca de se o anfitrião apresenta o estatuto de superhost ou não também deverá sempre ter um dos seguintes valores: verdadeiro ou falso, representados pelos valores 't' e 'f', respetivamente, através de uma restrição CHECK(isSuperHost in ('t', 'f')).

Como tal, no atributo superhost está presente a restrição NOT NULL e caso esta informação não seja fornecida deverá ser por DEFAULT falso, representado por 'f'.

A informação sobre quando o anfitrião começou a hospedar tem de estar presente, bem como o seu email, dado que não tem a possibilidade de se inscrever na plataforma sem um. Ambos os atributos devem ter uma restrição NOT NULL associada.

As taxas de resposta e de aceitação são ambas representadas por uma percentagem representada por um float entre 0.0 e 1.0. Ambos os atributos apresentam a restrição CHECK(0.0 <= atributo <= 1.0).

**Guest:** Cada hóspede é identificado pelo seu ID, dado que não pode haver dois hóspedes com o mesmo ID, pelo que guestID apresenta a restrição PRIMARY KEY.

Adicionalmente, também não pode haver dois hóspedes com o mesmo URL da foto de perfil. O atributo guestPictureURL é UNIQUE.

Cada hóspede tem um nome. guestName é NOT NULL.

**Comment:** Cada comentário é identificado pelo seu ID, pelo que commentID apresenta a restrição PRIMARY KEY.

Um comentário não pode estar vazio e a sua data deve estar bem definida, por este motivo, os atributos commentContent e commentDate são NOT NULL.

Cada comentário é escrito por um hóspede identificado pelo seu ID. Está presente a restrição guestID REFERENCES Guest(guestID).

Cada comentário é direcionado a um alojamento identificado também pelo seu ID. Assim, está presente a restrição listingID REFERENCES Listing(listingID).

**ReviewScores:** Cada conjunto de classificações é identificado pelo seu ID, pelo que reviewID apresenta a restrição PRIMARY KEY.

Ao classificar um alojamento, os hóspedes podem escolher em que categorias específicas desejam atribuir uma classificação. Contudo, a classificação geral (rating) deve ser sempre atribuída. Como tal, o atributo rating é NOT NULL.

Cada classificação, representada por um float, tem uma pontuação de 0 a 5, que é assegurada através da restrição CHECK(0.0 <= todas\_as\_classificações <= 5.0).

Cada comentário é atribuído por um hóspede identificado pelo seu ID a um alojamento também identificado pelo seu ID. Estão presentes as restrições guestID REFERENCES Guest(guestID) e listingID REFERENCES Listing(listingID).

**Reservation:** Cada reserva é identificada pelo seu ID, pelo que reservationID apresenta a restrição PRIMARY KEY.

Para cada reserva a data do check-in, a data do check-out e o número de adultos, crianças e bebés deve estar bem definido. Assim, os atributos checkIn, checkOut, nrAdults, nrChildren, nrBabies são NOT NULL.

Além disso, o número de adultos, crianças e bebés a serem alojados deve ser representado por um inteiro maior que zero ou igual no caso de crianças e bebés. Estão presentes as restrições CHECK (nrAdults > 0) e CHECK (nrChildren/Babies >= 0).

Além disso, a data do check-in deve ser anterior à data do check-out, pelo que está presente a restrição CHECK(checkIn < checkOut).

Cada reserva é feita por um hóspede identificado pelo seu ID para um alojamento também identificado pelo seu ID. Estão, portanto, presentes as restrições guestID REFERENCES Guest(guestID) e listingID REFERENCES Listing(listingID).

**Location:** A localização de cada alojamento é definida pela sua latitude e longitude, dado que as coordenadas de um local são únicas. latitude e longitude constituem uma chave primária composta. Está presente a restrição PRIMARY KEY(latitude, longitude).

A medição destes dois atributos é feita através de graus. A latitude deve variar entre -90° e 90° e a longitude entre -180° e 180°. Assim, para estes atributos deverão ser utilizadas as restrições CHECK (-90.0 <= latitude <= 90.0) e CHECK (-180.0 <= longitude <= 180.0).

A localização de cada alojamento pertence a um bairro identificado pelo seu ID. Está, portanto, presente a restrição neighbourhoodID REFERENCES Neighborhood(neighbourhoodID).

**Neighbourhood:** Cada bairro é definido pelo seu ID, logo o atributo neighbourhoodID apresenta uma restrição PRIMARY KEY.

Cada bairro tem de ter um nome. neighbourhoodName é NOT NULL.

Cada bairro pertence a um condado (neighbourhood group) identificado pelo seu ID. Assim, está presente a restrição neighbourhoodGroupID REFERENCES NeighbourhoodGroup(neighbourhoodGroupID).

**NeighbourhoodGroup:** Cada grupo de bairros é definido pelo seu ID, logo o atributo neighbourhoodGroupID apresenta uma restrição PRIMARY KEY.

Cada grupo de bairros tem de ter um nome. neighbourhoodName é NOT NULL.

**AmenityType:** Cada comodidade é identificada pelo seu ID, pelo que amenityID apresenta uma restrição PRIMARY KEY.

Cada tipo de comodidade tem um nome. amenityType é NOT NULL.

**hasAmenity:** A associação entre um alojamento e as comodidades que este possui é identificada pelo ID do alojamento e da comodidade. Assim, listingID e amenityID constituem a chave primária da relação, tendo-se então a restrição PRIMARY KEY(listingID, amenityID).

Esta chave primária é constituída por elementos de outras relações que constituem chaves estrangeiras. Deste modo, estão presentes as restrições listingID REFERENCES Listing(listingID) e amenityID REFERENCES Amenity(amenityID).

**Season:** Cada época é identificada pelo seu ID, pelo que seasonID apresenta a restrição PRIMARY KEY.

As datas de início e fim de cada época devem estar bem definidas. Assim, startDate e endDate são ambos NOT NULL.

Além disso, a data de início de uma época tem de ocorrer antes da data de fim da mesma. Deste modo, encontra-se presente a restrição CHECK (startDate < endDate).

**Pricing:** Para cada época do ano, cada alojamento tem um preço por noite identificado pelo ID do alojamento e pelo ID da época. Assim, estes dois atributos constituem a chave primária da relação. Está presente a restrição PRIMARY KEY(listingID, seasonID).

Esta chave primária é constituída por elementos de outras relações que constituem chaves estrangeiras. Deste modo, estão presentes as restrições listingID REFERENCES Listing(listingID) e seasonID REFERENCES Season(seasonID). Além disso, o preço por noite tem de estar bem definido, pelo que se encontra presente a restrição NOT NULL, e deve ser maior que zero, pelo que apresenta a restrição CHECK (pricePerNight > 0).

## 7. Interrogações À Base de Dados

- 1) Lista de hóspedes que se encontram atualmente hospedados num alojamento, os seus nomes, a data em que o check-in foi realizado e a data em que terão o seu check-out.
- 2) Cálculo da média da classificação geral de um alojamento e dos seus parâmetros adicionais, tais como limpeza, check-in, comunicação e localização para todos os alojamentos que possuem essa informação.
- 3) Preços para os alojamentos disponíveis no verão de 2022, capacidade máxima desses alojamentos e o preço por pessoa, assumindo que é feita uma reserva que vai de acordo com a capacidade máxima de pessoas que o alojamento permite.
- 4) Lista dos hóspedes que só ficam hospedados em alojamentos de superhosts, isto é, que não têm nenhum registo de reservas para alojamentos que não sejam hospedados por superhosts.
- 5) Qual é a reserva mais antiga, quais foram as datas de check-in e de check-out, em que alojamento foi realizada e quem foi o anfitrião.
- 6) Classificação (rating) média dos alojamentos de cada bairro. Esta informação pode ser relevante para quem quiser saber quais as zonas da cidade que têm, de uma modo geral, melhores alojamentos.
- 7) Lista das reservas com check-in amanhã. O alojamento que irá receber os hóspedes, o hóspede que efetuou a reserva e informação sobre a reserva: número de adultos, crianças e bebés.
- 8) Número de reservas antes e depois da pandemia, mais especificamente em 2019, em 2020 e em 2021.
- 9) Listagem dos alojamentos que possuem todas as 5 amenities consideradas essenciais num alojamento: água quente, wi-fi, extintor de incêndio, frigorífico e kit de primeiros socorros.
- 10) Enumeração de todos os hóspedes considerados inativos, ou seja, aqueles que nunca fizeram uma reserva ou que a última reserva que efetuaram foi há mais de 3 anos.

## 8.Gatilhos

- 1) Assegura que uma nova reserva apenas contém um número de hóspedes (adultos, crianças e bebés) menor ou igual à capacidade do alojamento, rejeitando reservas que violem esta restrição.
- 2) Impede a remoção de um alojamento local da base de dados caso ainda existam reservas por realizar.
- 3) Assegura que uma nova reserva apenas possa ser feita se a data de check-in for posterior à data atual.

## 9. Avaliação da participação de cada elemento do grupo

Durante a realização do trabalho houve participação ativa de todos os elementos do grupo e verificou-se um grande espírito de entreajuda.

Nesta última entrega, em específico, dividimos as tarefas da forma mais equilibrada possível. Um de nós ficou responsável pela implementação dos triggers e os restantes por 5 queries cada um. No entanto, todos contribuíram na finalização do relatório e com ideias.

	Entrega 1	Entrega 2	Entrega 3
Anete Pereira	33,3%	33,3%	33,3%
Isabel Amaral	33,3%	33,3%	33,3%
Tomás Pires	33,3%	33,3%	33,3%