# 

João Marcelo de Jesus Macedo

Matheus Reis de Souza Teixeirense

Matheus Sena Vasconcelos

Sistema hoteleiro - hospedar.com

Sistema de informação - 05 / 2020

Administração de Banco de Dados

Professor Roger Oliveira

# **Sumário**

[**Sumário**](#_a61v1ihi23fg) **2**

[**Sobre**](#_kokkecre4d4c) **3**

[Nome do projeto](#_92aqe02w6jik) 3

[Objetivo](#_r69h3k9sgutx) 3

[Como usar](#_foo1ydp6yuoz) 3

[Banco de dados utilizado](#_coeqz8ggtand) 3

[Links](#_tcj6n8vr5hrf) 3

[**Estrutura do projeto**](#_sh5ljstf751e) **4**

[Modelo de Entidade e Relacionamento (MER)](#_pza2ffodds7d) 4

[**Tabelas, índices e suas telas**](#_6qh8o0cmsb4n) **4**

[Tipo de pagamento](#_h5du95ibt37o) 4

[Quarto e tipo do quarto](#_u3u0lqjvjdm6) 6

[Hóspedes, funcionários e seus endereços](#_wvynqzfxy4vt) 9

[Hospedagem](#_c44hs374kesu) 15

[**Triggers**](#_2ecyqxehw4i1) **21**

[Validação dos dados](#_atv63rep0fua) 21

[Calculando o valor da hospedagem](#_tb0m0nmp24yb) 24

[**Views e os relatórios**](#_fqwi1njzqi7s) **24**

[Overview](#_9qj765hn7s5j) 24

[Detalhado](#_5kq4btz9h8oc) 26

[Rank](#_o43wnlpk76sj) 27

[Outros](#_lexgutv7k47l) 28

[**Estimativa da quantidade de linhas durante 1 ano**](#_w426auchuxxr) **28**

# **Sobre**

## Nome do projeto

Hospedar.com

## Objetivo

Desenvolver um sistema web para controle de atividades hoteleiras, onde seja possível o cadastro dos quartos e seus valores, formas de pagamento, funcionários atuantes e os hóspedes do hotel. Que seja possível, também, a disponibilização de relatórios, podendo ser em arquivos, telas ou dashboards. Além disso, oferecer ao usuário do sistema um controle das reservas das hospedagens mais simples, rápida e consistente, trazendo uma visualização prévia dos quartos e hóspedes disponíveis.

## Como usar

Depois de configurar e subir o sistema em um servidor, é necessário primeiro cadastrar seus funcionários, quartos, valores das diárias e formas de pagamentos disponíveis.

Quando o sistema estiver populado com esses dados básico, já pode-se usar em seu hotel. Basta cadastrar os novos hóspedes que chegarem em seu hotel e vincular esses dados à uma hospedagem.

## Banco de dados utilizado

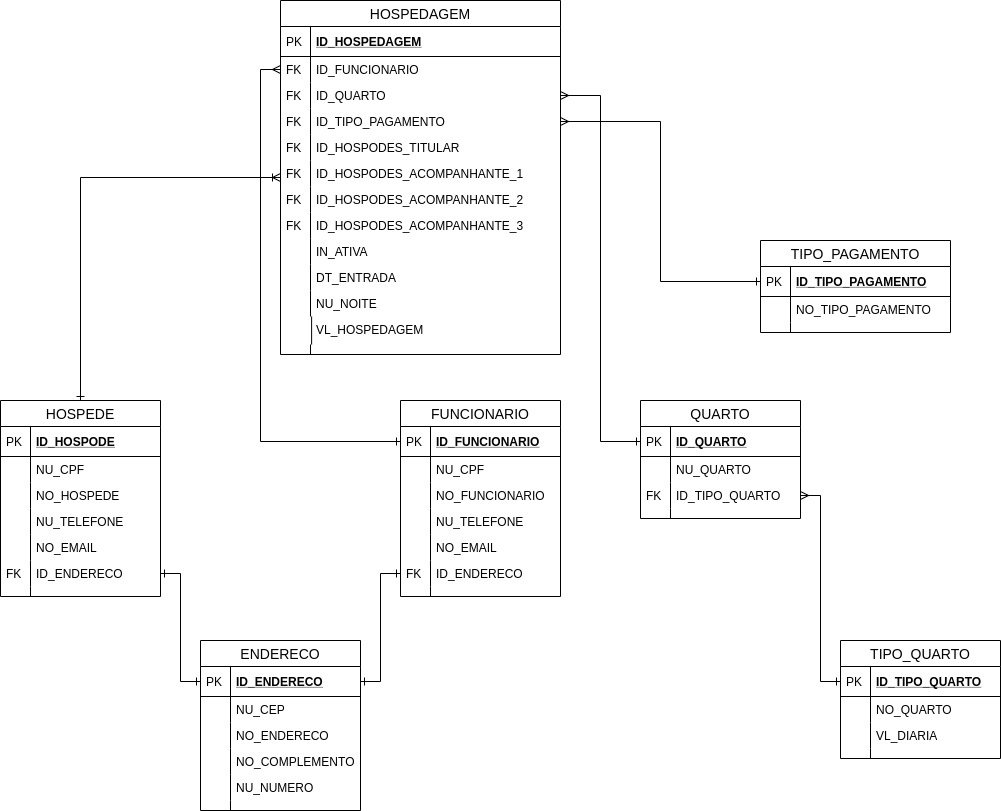
SQLite3

## Links

O código de todo o projeto, desde o Banco de Dados ao back-end em Python3 e o front-end em HTML e Jinja2, podem ser visualizados no repositório público do GitHub <https://github.com/senavs/website-hosting>.

# **Estrutura do projeto**

## Modelo de Entidade e Relacionamento (MER)

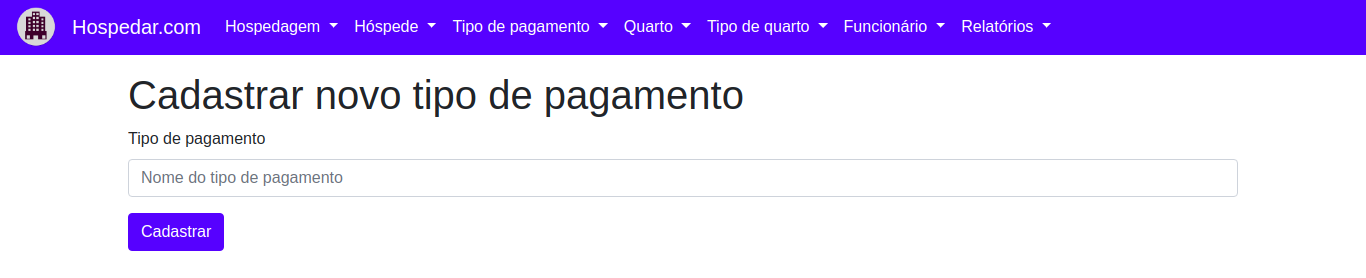
MER do hospedar.com

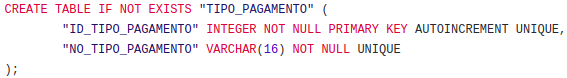
# Tabelas, índices e suas telas

Para começar os scripts de criação das tabelas, devemos começar pelas entidades que não possuem chaves estrangeiras, ou seja, as tabelas de domínio.

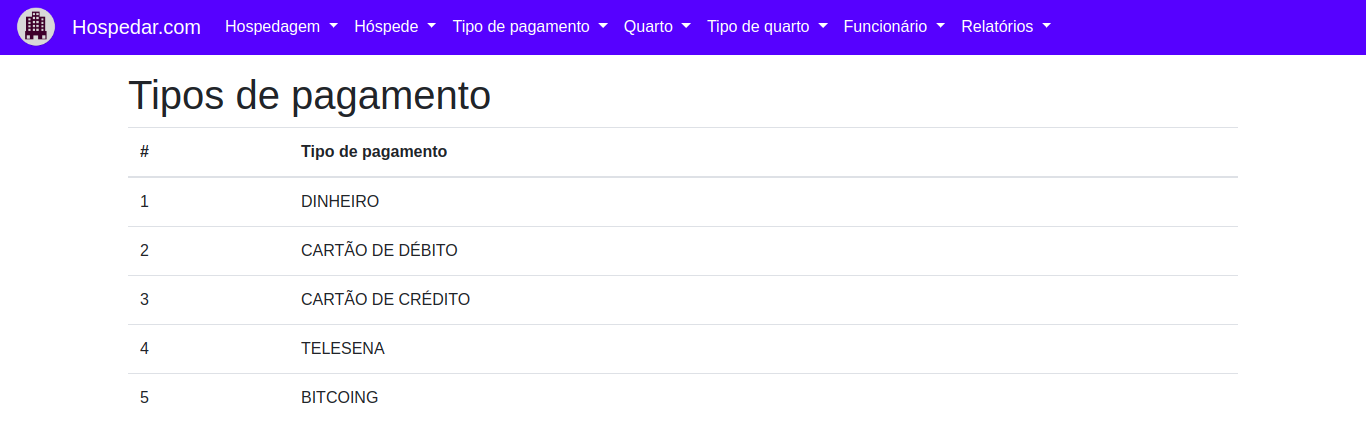
## Tipo de pagamento

Tipo de pagamento é uma tabela assim. Criada simplesmente para armazenar os possíveis tipos de pagamento que o hotel pode aceitar.



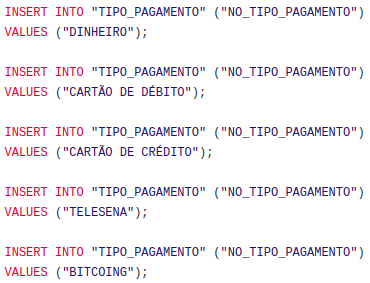


Como o próprio nome do tipo de pagamento é único, igual à chave primária, então é possível transformá-la em índice para facilitar a consulta.

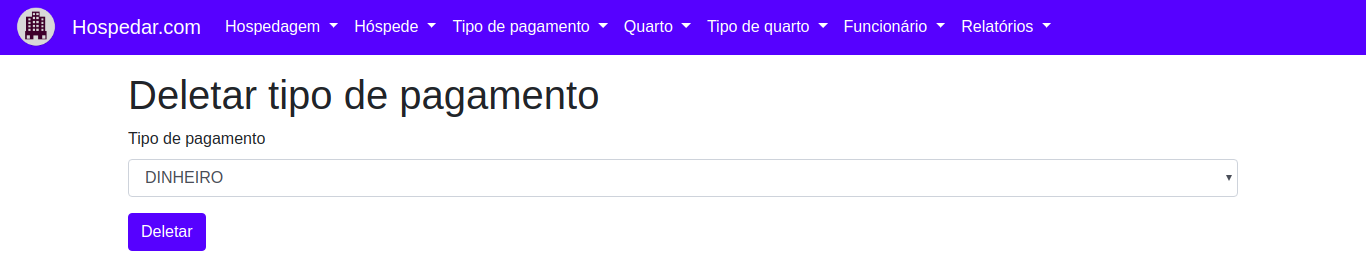




Para popular a tabela dos tipos de pagamento, o seguinte código DML foi utilizado:



Além disso, é possível deletar um tipo de pagamento previamente cadastrado:



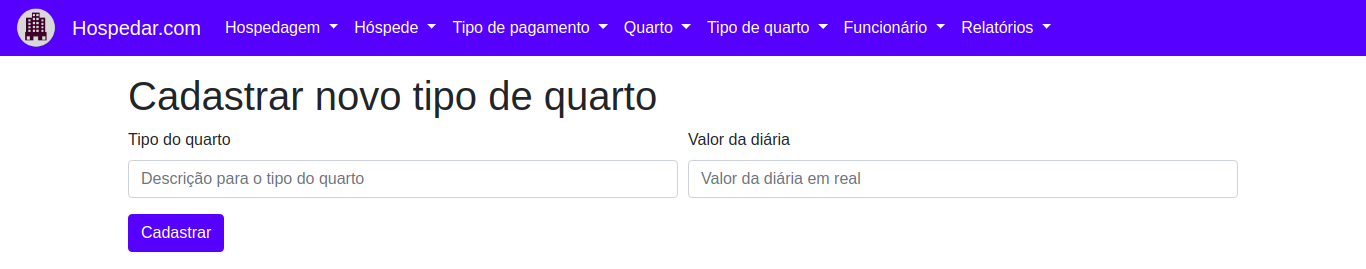


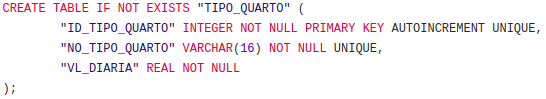
Para mostrar todos os tipos de pagamentos disponíveis para deletar, é necessário fazer uma consulta que busque todos os dados dessa tabela.



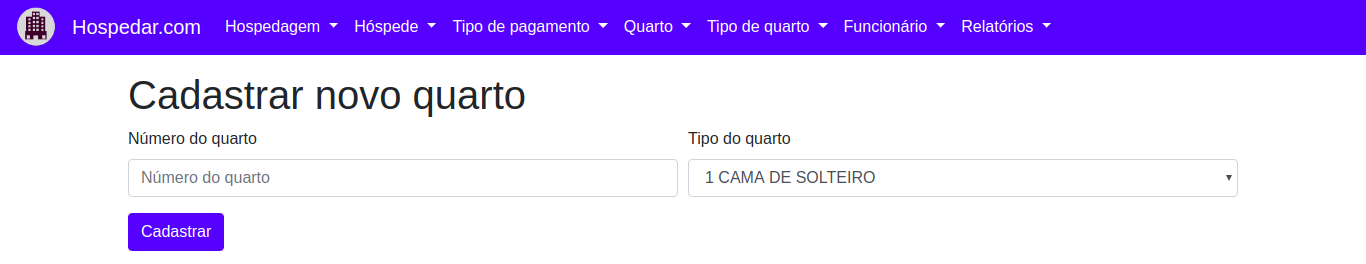
## Quarto e tipo do quarto

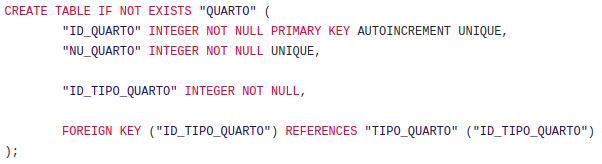
Para cadastro do quarto é preciso de duas tabelas. A primeira é a tabela domínio para armazenar os tipos de quarto e seus respectivos valores.





A outra tabela seria o próprio quarto, sendo necessário armazenar o número do quarto e vinculá-lo com o tipo do quarto através da chave estrangeira.



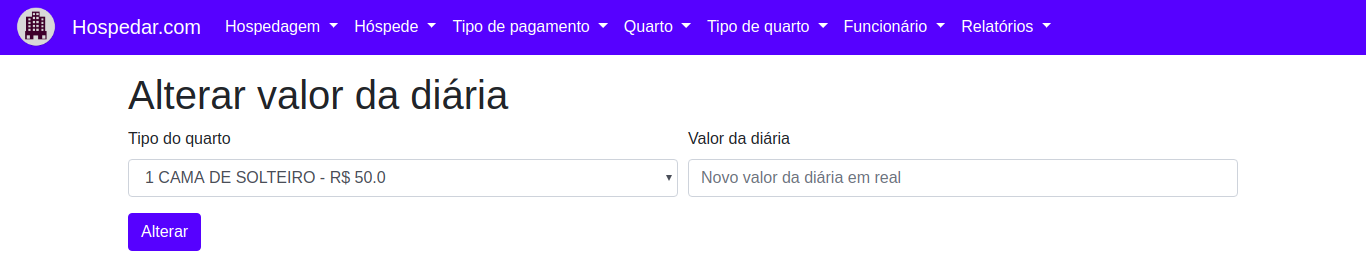


Do mesmo jeito do tipo de pagamento, a tabela quarto possui a coluna Número do Quarto que contém valores únicos, podendo ser criados índices.

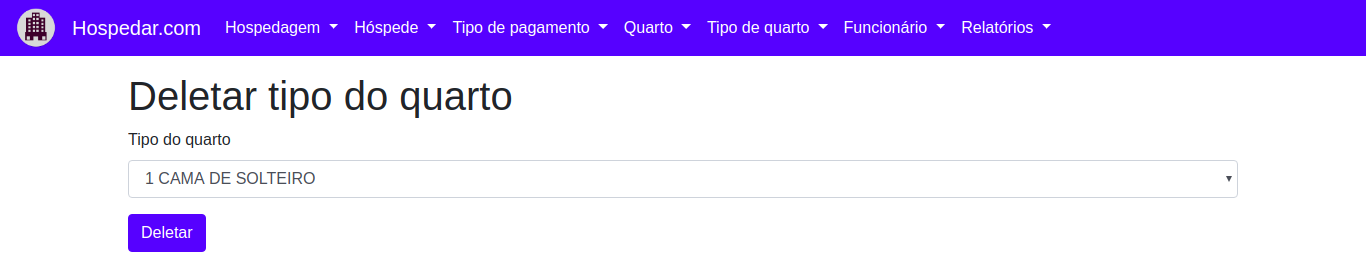




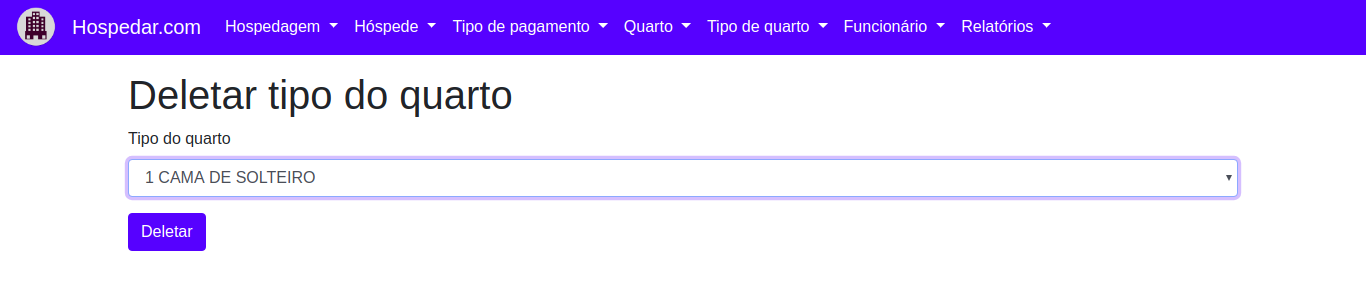
Além dessas telas, ainda possui telas para alterar o valor da diária e deletar o quarto e o tipo de quarto.





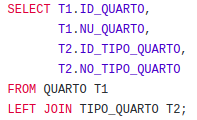






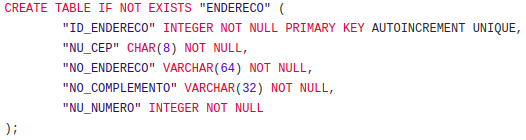


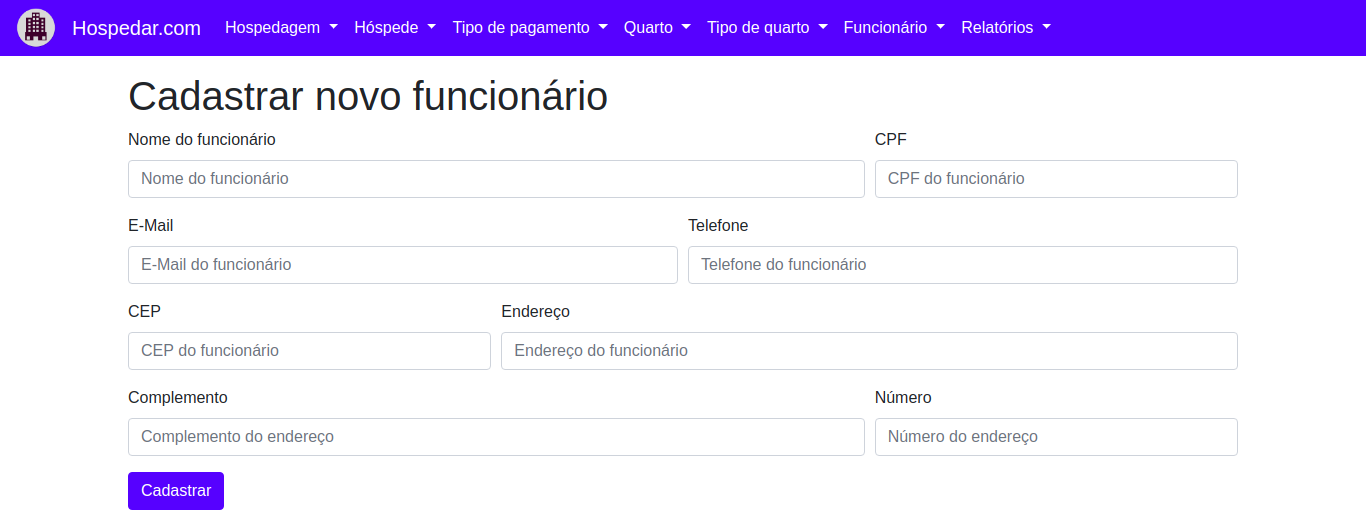
Nessas últimas telas acima, é necessário fazer uma consulta para buscar todos os dados presentes nas tabelas quarto e tipo do quarto e preencher o dropdowncom os quartos e tipos disponíveis.

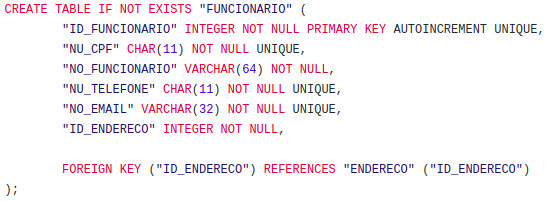


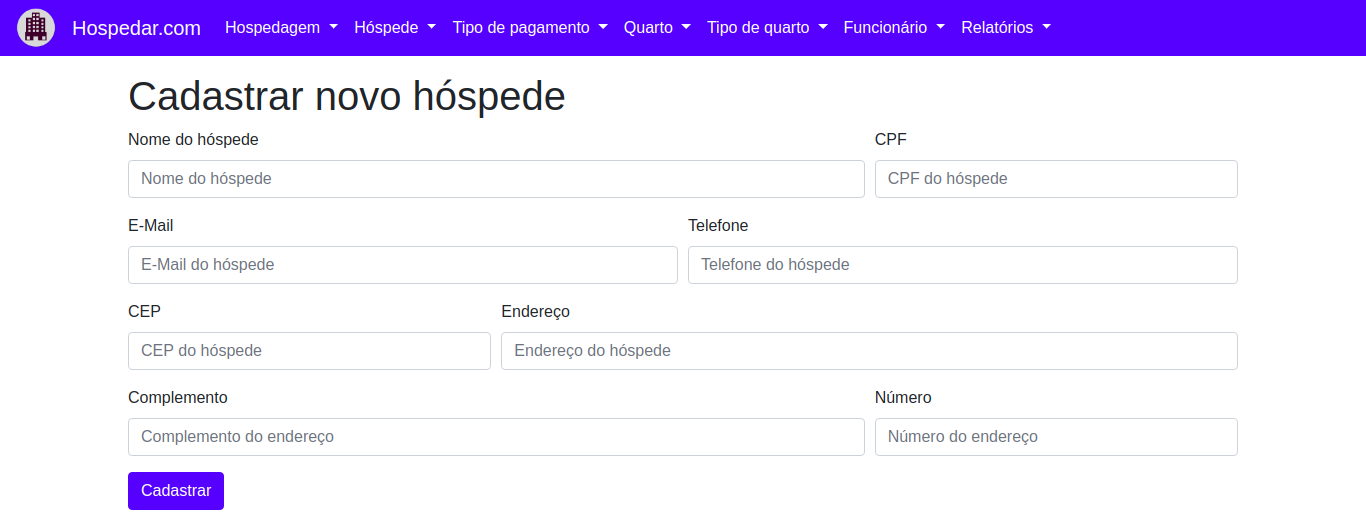
## Hóspedes, funcionários e seus endereços

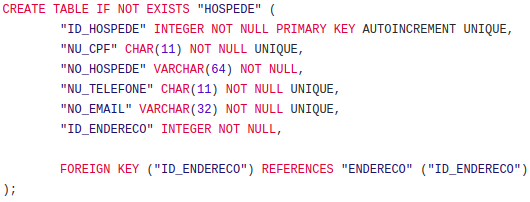
No MER, observa-se que os Hóspedes e Funcionários estão conectados à mesma tabela: Endereço. Então, já que essas duas tabelas possuem chaves estrangeiras da tabela endereço, é necessário construí-la primeiro e, logo em seguida, as tabelas de funcionário e hóspede.



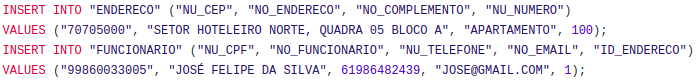


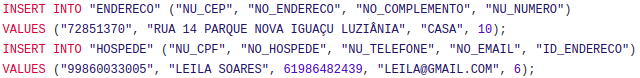




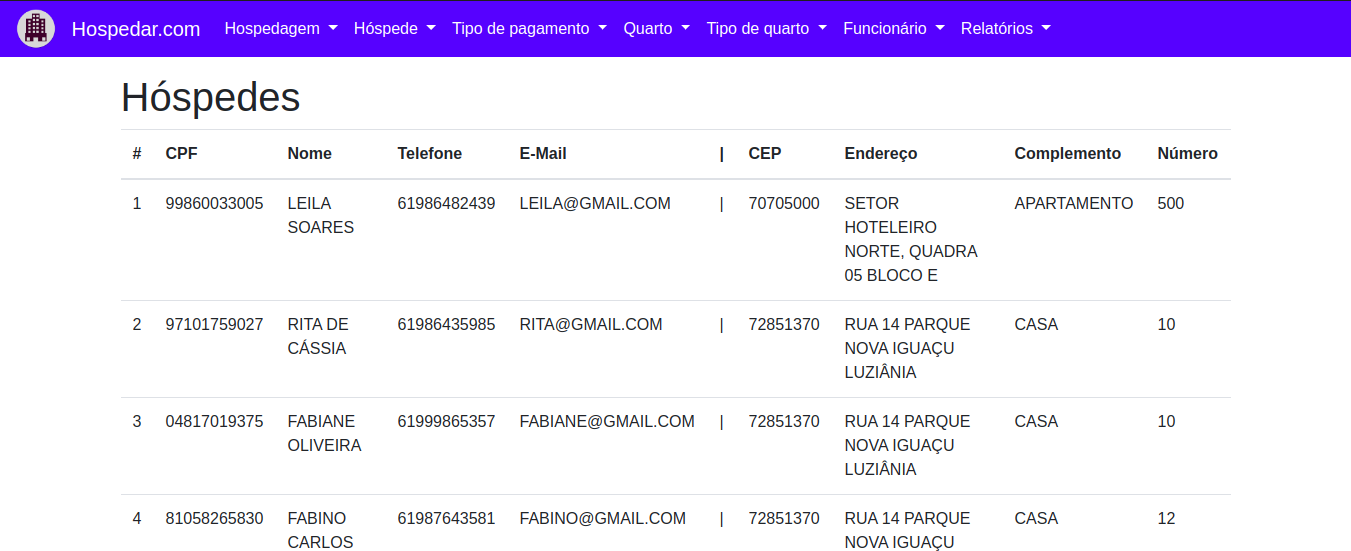


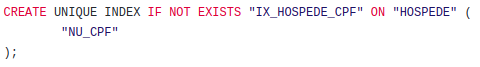
Nessas duas telas, em cada uma, é necessário fazer duas inserções no banco de dados. Primeiramente, é necessário cadastrar o endereço do hóspede/funcionário. Caso a inserção ocorra com sucesso, podemos inserir a pessoa passando a chave estrangeira do endereço. Caso uma das duas inserções falhem, é necessário fazer o rollback.

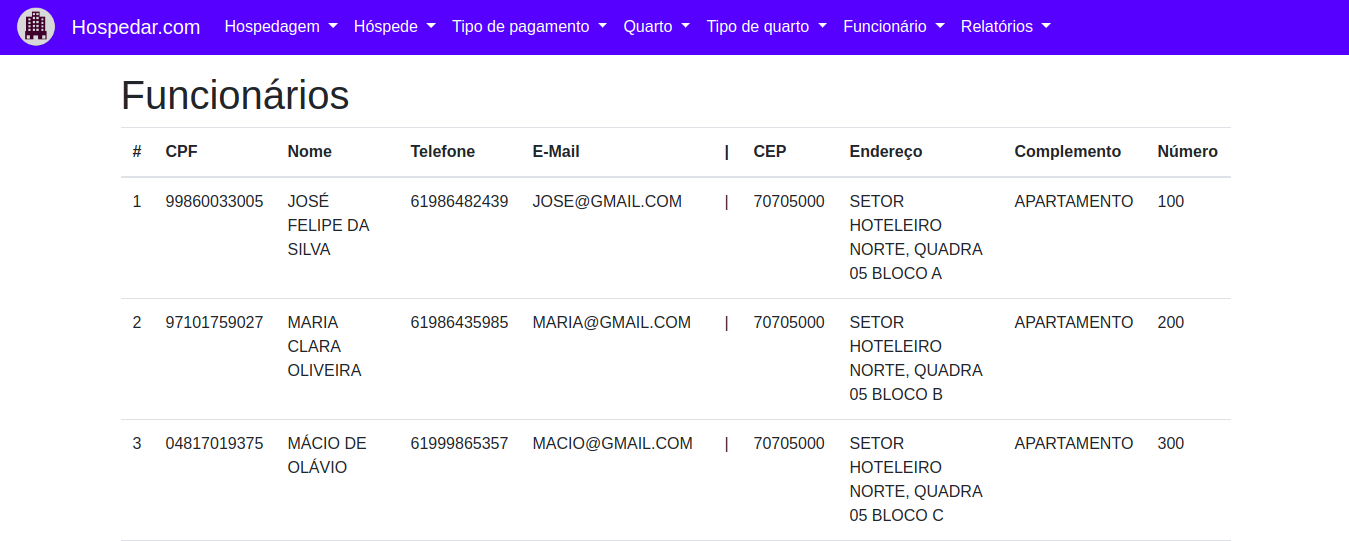


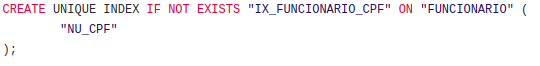


Do mesmo jeito das outras tabelas listadas acima, as tabelas Hóspede e Funcionários possuem colunas de valores únicos, como por exemplo o CPF. Então, para uma maior velocidade das consultas à essas tabelas, podemos criar índices.

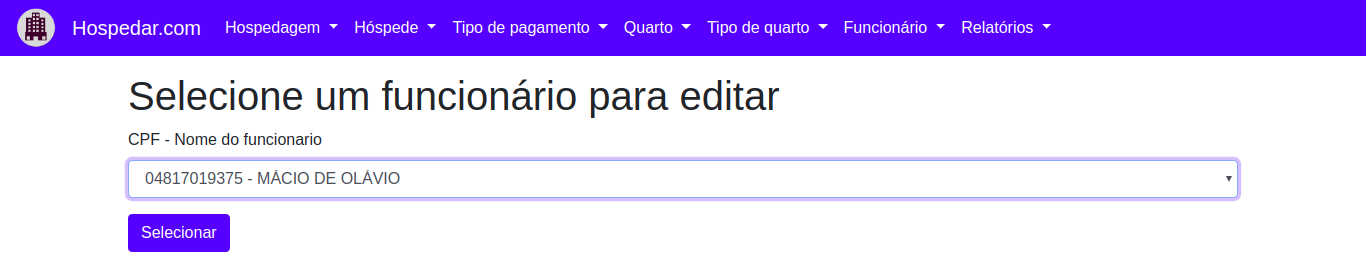


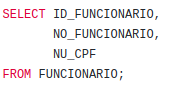


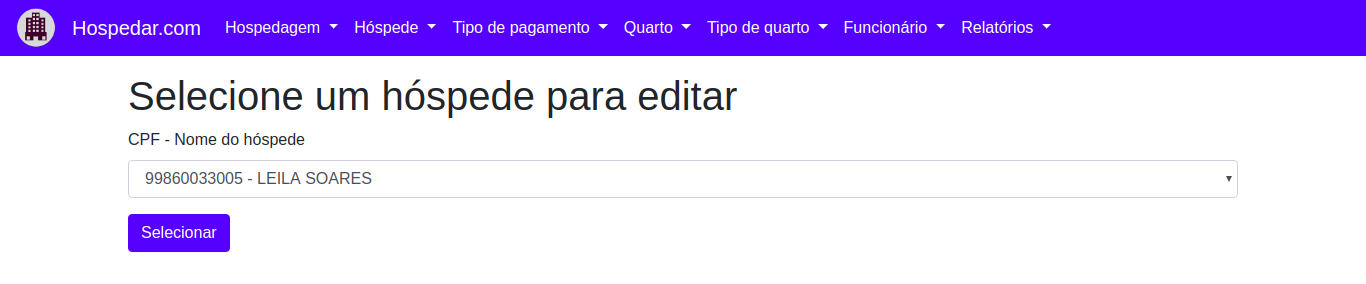


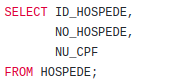


Para alterar os dados dos funcionários e dos hóspedes, é necessário também alterar a tabela de endereço. Primeiramente, Listamos todos as pessoas disponíveis para essa alteração, fazer uma busca de todos os registros dentro do banco de dados.

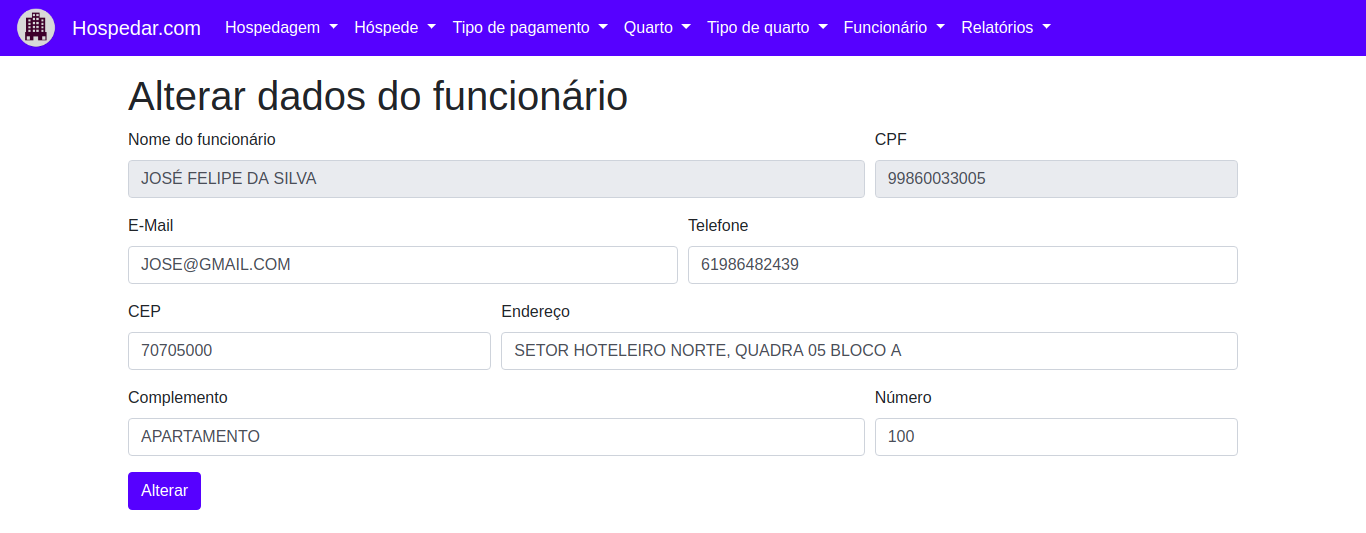


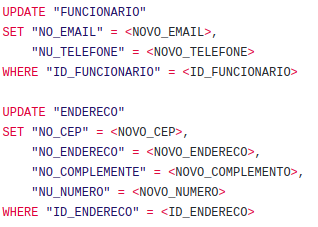


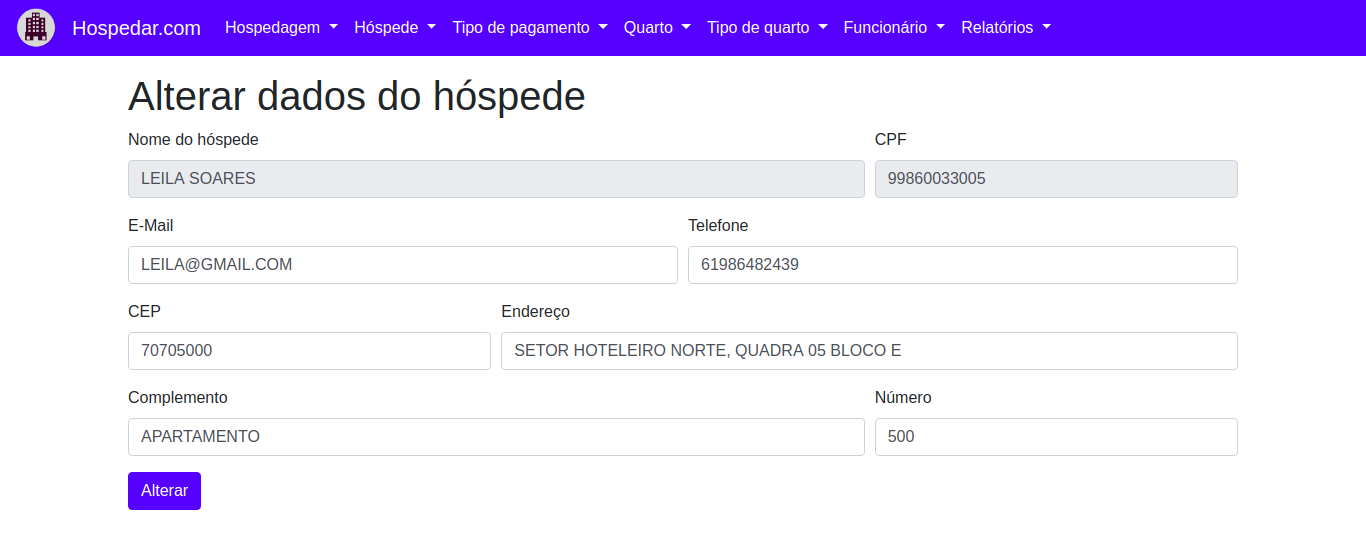


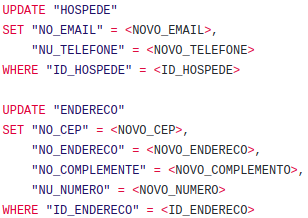


Depois de selecionado, outra busca é realizada, porém juntando com a tabela de endereço e filtrando pelo hóspede/funcionário selecionado. Com o resultado da consulta é possível construir uma tela que disponibiliza esses dados consultados e preenchidos. Caso o usuário selecione o botão alterar, todos os dados são alterados.

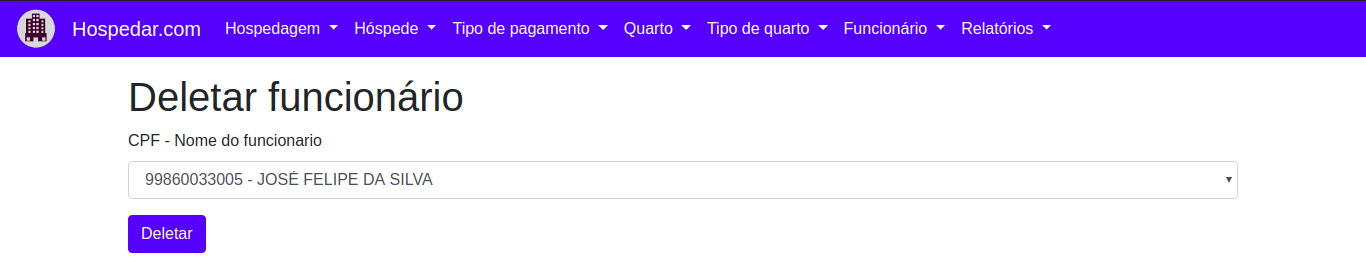


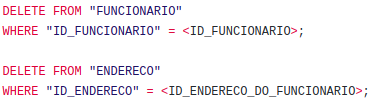


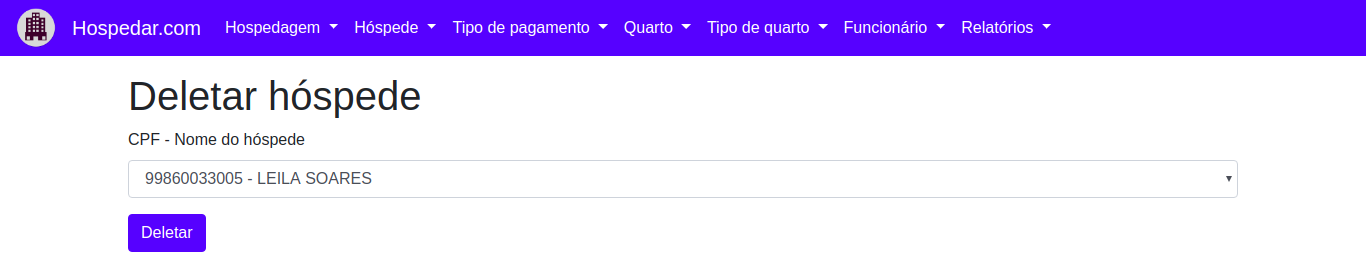


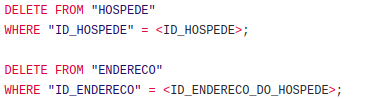


Também é possível deletar as pessoas do sistema. Para deletar um hóspede/funcionário, também é necessário deletar seu endereço.



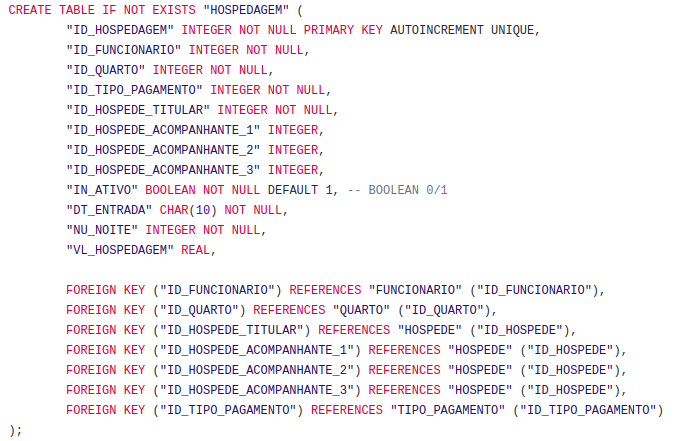




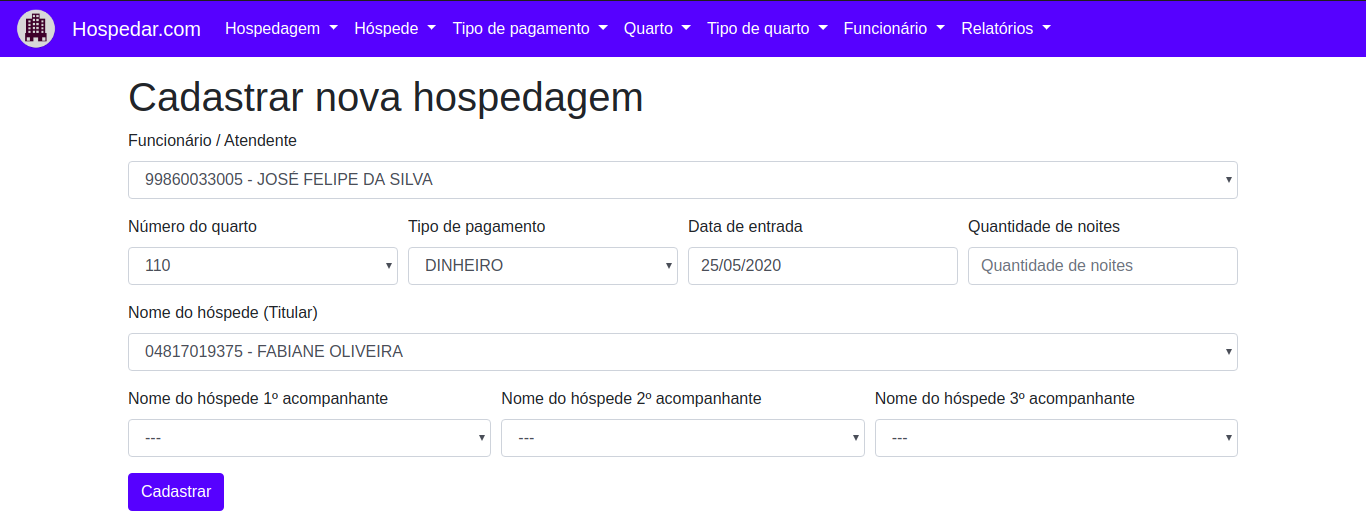


## Hospedagem

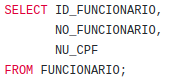
A tabela Hospedagem, que é o centro da aplicação, armazena todas as chaves primárias das outras tabelas. É necessário guardar a chave primária do funcionário que efetuou a reserva, a chave primária do quarto e do tipo de pagamento selecionado e as chaves primárias dos hóspedes que estarão na reserva. Os hóspedes acompanhantes não são obrigatórios, pois, para reservar um quarto, basta um hóspede. Além disso, é necessário armazenar se a atual hospedagem está ativa (o valor default é 1), a data que a reserva foi criada e quantas noites os hóspedes vão ficar. O valor da hospedagem não é necessário preencher, pois será preenchido pela [trigger mais para frente.](#_tb0m0nmp24yb)



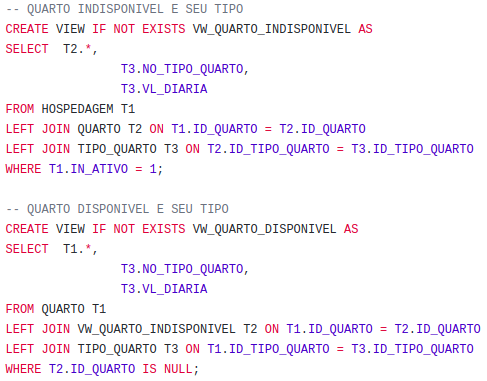
Para a tela de cadastrar hospedagem, algumas consultas são realizadas para preencher os dropdown.



Primeiramente, trazer todos os funcionários cadastrado no sistema:



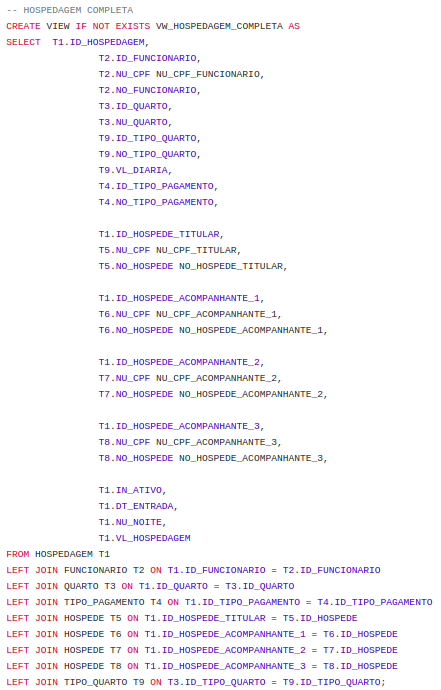
Depois, é preciso listar somente os quartos disponíveis, ou seja, quartos que não estão com a hospedagem aiva. Para isso, podemos criar uma view, pois usaremos essa mesma consulta para a criação da dashboard do sistema (espécie de relatório). Primeiro criamos uma view para listar todos os quarto indisponíveis, para aí sim, buscar os quartos disponíveis através da diferença.

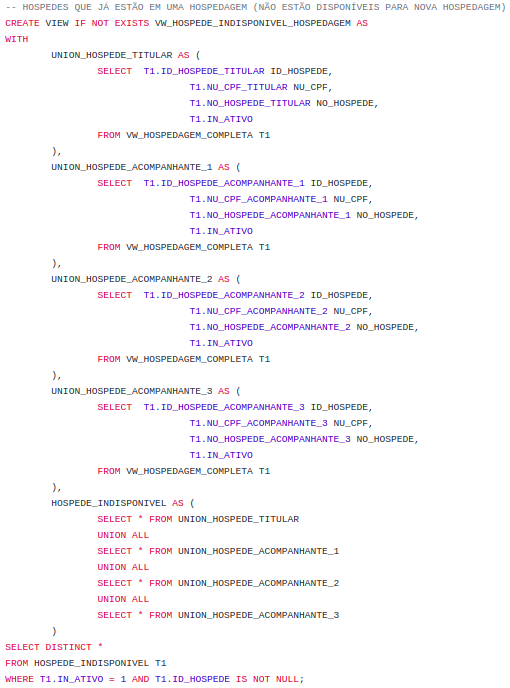


Além disso, listar todos os tipos de pagamentos disponíveis:

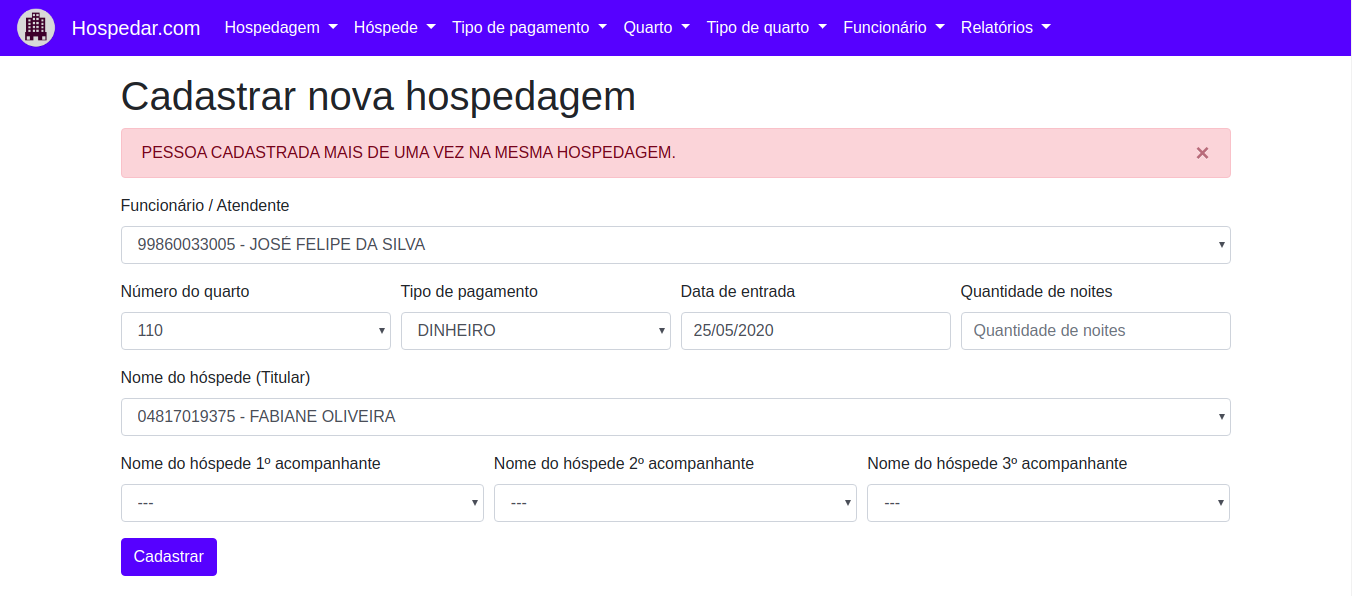


A data é preenchida automaticamente com a data de hoje, porém o usuário pode alterar. Depois de digitar a quantidade de noites que os hóspedes irão ficar, o usuário pode selecionar os hóspedes. Para isso, o sistema deve mostrar os hóspedes que não estão em hospedagens ativas. As seguintes views fazem esse papel:

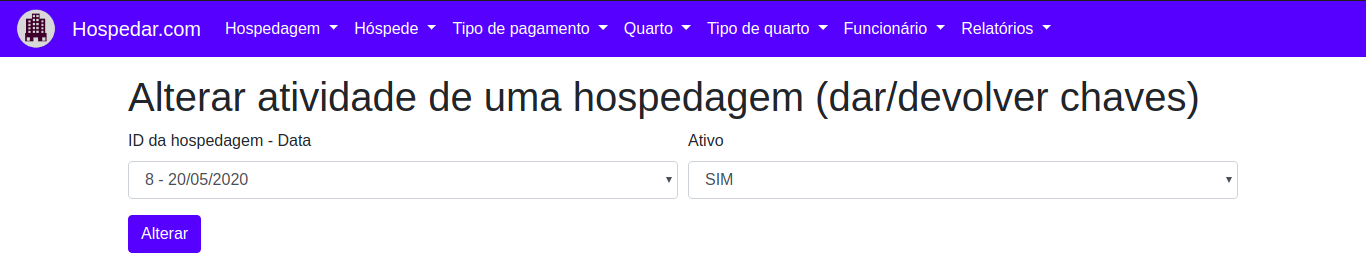




**Obs:** o próprio sistema verifica se o um mesmo hóspede disponível foi utilizado mais de uma vez na mesma hospedagem através das trigger (listada mais adiante).



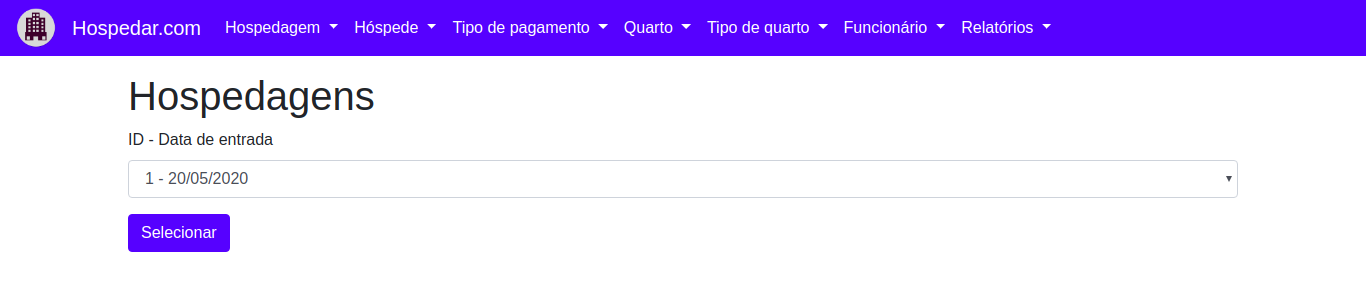
Para a hospedagem, a alteração é somente se ela está ativa ou desativa. Para isso, uma tela que mostra todas as hospedagens e seus identificadores faz a seguinte consulta:

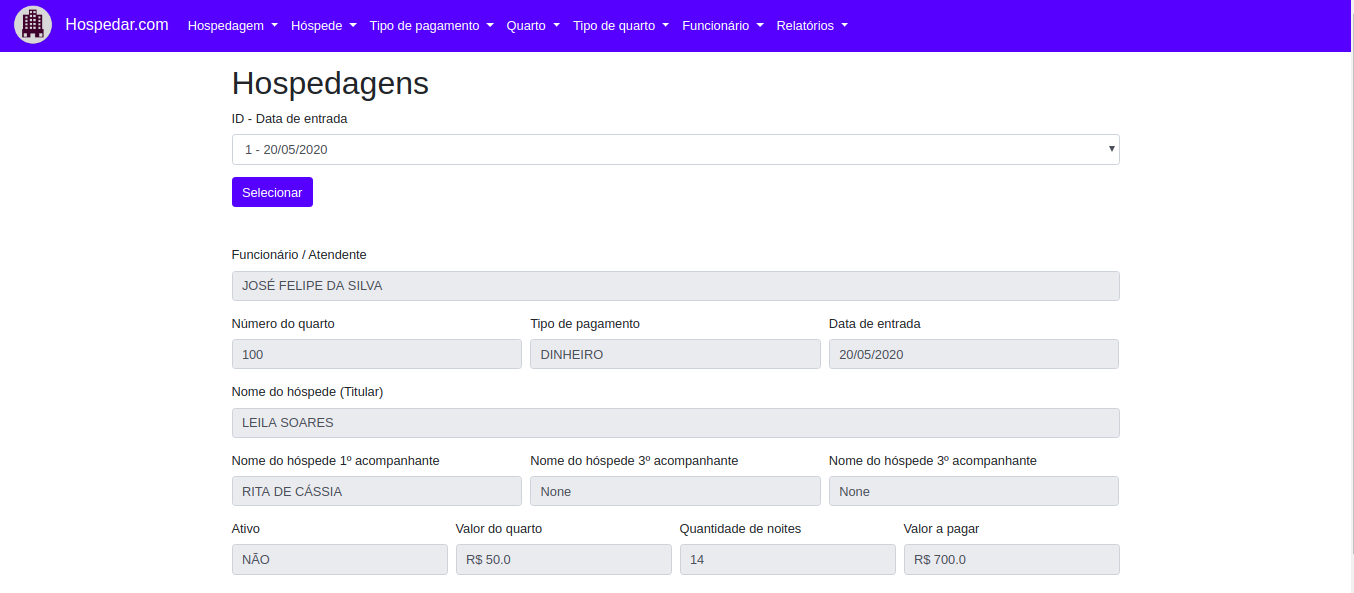




Depois, o valor é alterado com utilizando a cláusula UPDATE.

Para verificar dados da hospedagem já cadastrada anteriormente, a mesma consulta é utilizada para mostrar as hospedagens disponíveis. Após selecionado, a seguinte tela é gerada a partir das views de hospedagem construídas anteriormente.

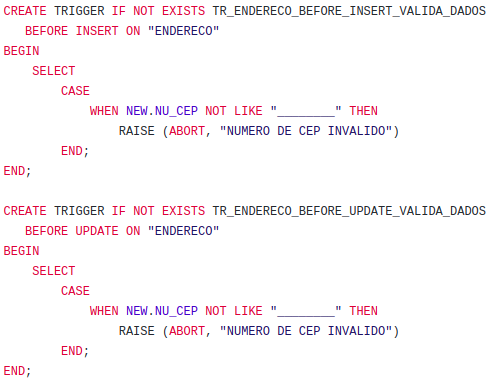




# Triggers

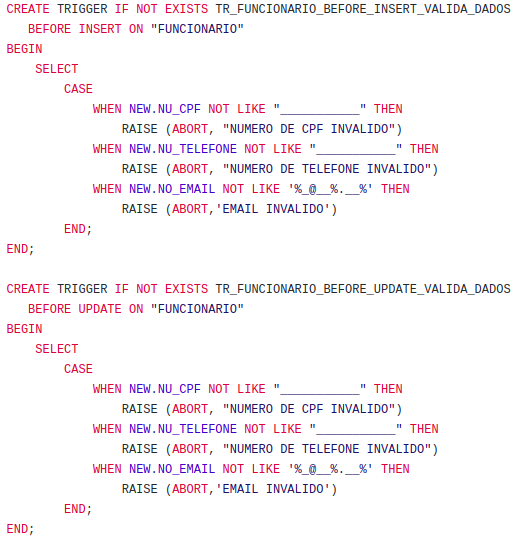
## Validação dos dados

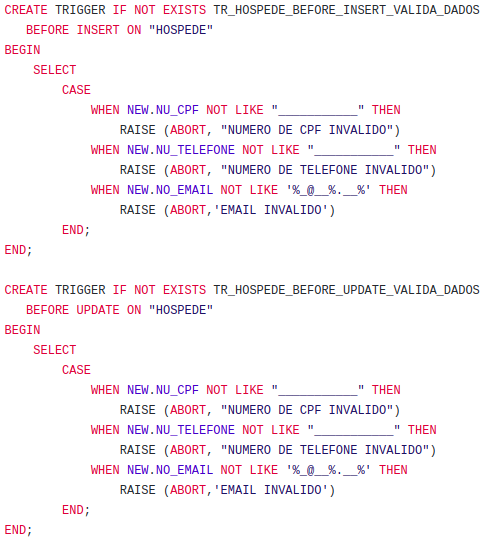
Em algumas colunas, é necessário verificar o tipo ou o formato dos dados que estão entrando nas tabelas. Por exemplo, na tabela *Endereço* o campo CEP não pode aceitar qualquer formato, é necessário que seja somente um campo texto de 8 dígitos de números. Podemos garantir essa regra com as seguintes triggers.



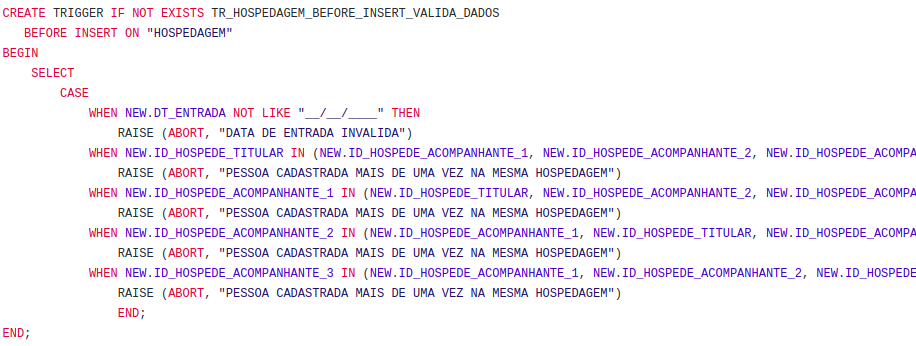
Duas triggers foram criadas para garantir a mesma lógica durante a inserção e durante a alteração.

O mesma acontece nas tabelas *Funcionários* e *Hóspedes*. Devemos garantir que os dados de CPF, E-Mail e telefone sejam preenchidos corretamente.



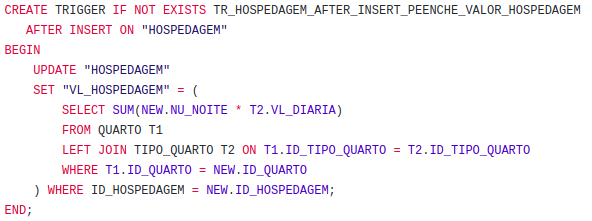


Para a tabela hospedagem não é diferente, é preciso fazer as validações dos dados. Porém, a única validação necessária é verificar se na mesma hospedagem o hóspede não está sendo registrado duas vezes.



## Calculando o valor da hospedagem

Outra trigger necessária na tabela hospedagem é para calcular o valor toda da hospedagem. A seguinte trigger verifica a quantidade de noites que os hóspedes vão ficar e multiplica pelo valor da diária do quarto selecionado. Após calculado, o valor é inserido na tabela.



# Views e os relatórios

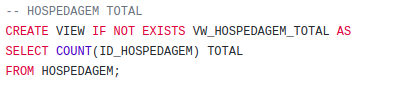
Na tela inicial do sistema possuem algumas tabelas com dados. Essa é a dashboard do sistema. **Cada cada valor marcado em amarelo nas imagens abaixo é uma consulta ou view**.

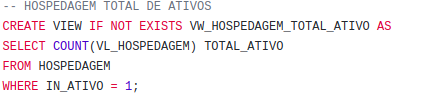
## Overview

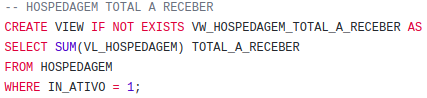
Primeiramente, temos a seção Overview. Nela é possível ter uma noção geral do hotel.

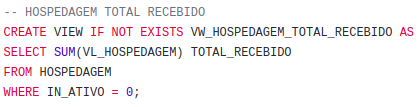


Para os dados da hospedagem, são listados: Total de hospedagem cadastradas, hospedagem ativas, total a receber em real (hospedagens ativas) e total já recebido em real (hospedagens desativadas). As seguintes fazem esse papel:

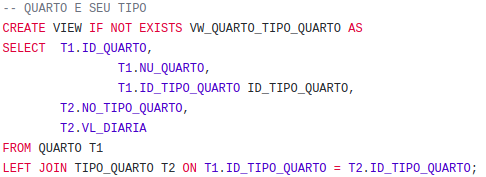


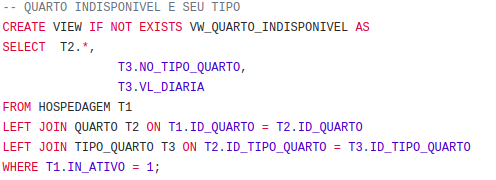


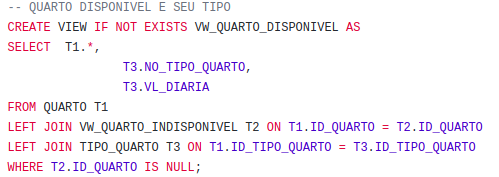




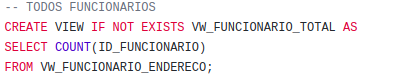
Ainda na seção Overview, temos um bloco para visualização dos quartos disponíveis. Novamente, views foram criadas para buscar esses dados do banco de dados.

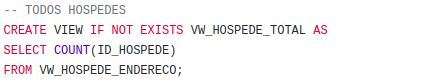






Para a dashboard dos hóspedes e funcionários não é diferente.

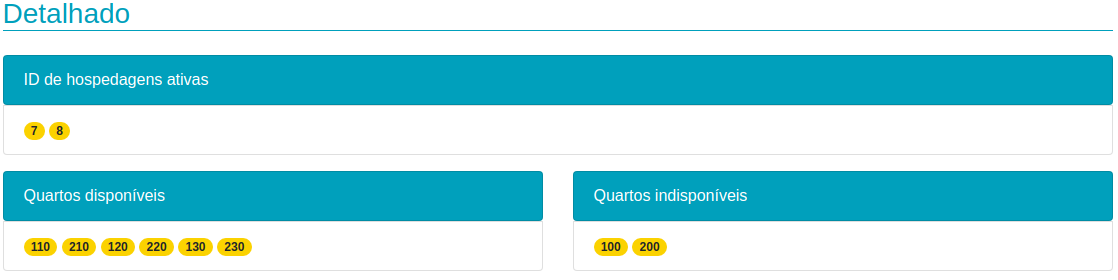




**Obs:** O banco de dados utilizado (SQLite3), em sua sintaxe, não é necessário utilizar a cláusula GROUP BY juntamente com a cláusula COUNT, quando selecionado apenas uma tabela.

## Detalhado

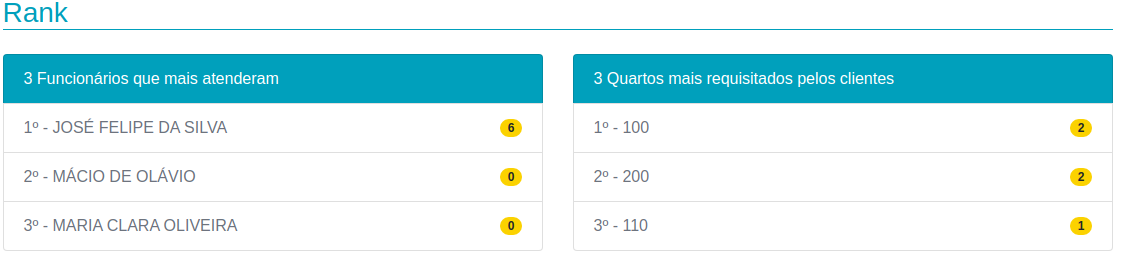
Na seção Detalhado da dashboard são apresentados dados das hospedagens e dos quartos, mostrando as chaves primárias (identificadores) das hospedagens ativas e os números dos quartos disponíveis e indisponíveis.



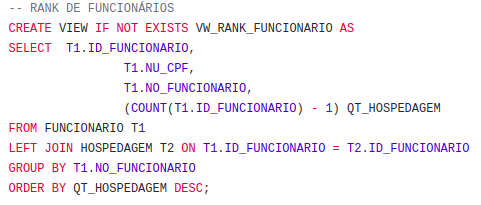
As views utilizadas nessa seção foram as mesma da [Overview](#_9qj765hn7s5j), porém, sem a contagem.

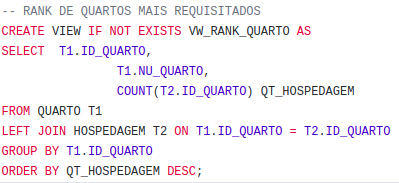
## Rank

Além disso, a dashboard traz uma seção para ranquear os funcionários que mais atenderam e os quartos mais solicitados.



Esses resultados podem ser obtidos através das seguintes views:





## Outros

O sistema também disponibiliza relatórios detalhados de cada entidade do sistema, conforme pode ser visto na seção [Tabelas, índices e views](#_6qh8o0cmsb4n) do documento.

# Estimativa da quantidade de linhas durante 1 ano

Como o sistema foi projetado para controle das hospedagem de um hotel, as tabelas que mais poderão receber dados seriam a de Hóspedes e de Hospedagem. Considerando que o hotel receba, em um mês, 30 novos hóspedes e registre 55 hospedagens, as tabelas de Hóspedes e Hospedagem estariam, ao final de um ano, com 360 e 660 registro, respectivamente.

Então, um banco de dados como por exemplo o MySQL com um plano de hospedagem de dados básico, conseguiria atender o sistema por, pelo menos, 10 anos.