Centro Federal de Educação Tecnológica de Minas Gerais

Bacharelado em Engenharia da Computação

Professora: Evandrino G. Barros

Sistema de Avaliação de Restaurantes

Realizado por:
Gabriel Torres Talim
João Victor Sfredo de Castro
Lara Santos Rossetti
Paulo Renato Souza Magalhães

Belo Horizonte
2023

Sumário

Descrição textual do banco de dados

Nome do Banco de Dados: Sistema de Avaliação de Restaurantes (BD-Restaurante)

O Banco de Dados "Sistema de Avaliação de Restaurantes" (BD-Restaurante) foi projetado para oferecer aos usuários uma plataforma interativa e informativa para avaliar e descobrir restaurantes, compartilhar experiências gastronômicas e tomar decisões informadas ao escolher um local para refeições. Este banco de dados é projetado para atender às necessidades dos amantes da culinária, permitindo que os usuários avaliem restaurantes, compartilhem suas opiniões, encontrem novos lugares para comer e interajam com outros apreciadores de gastronomia.

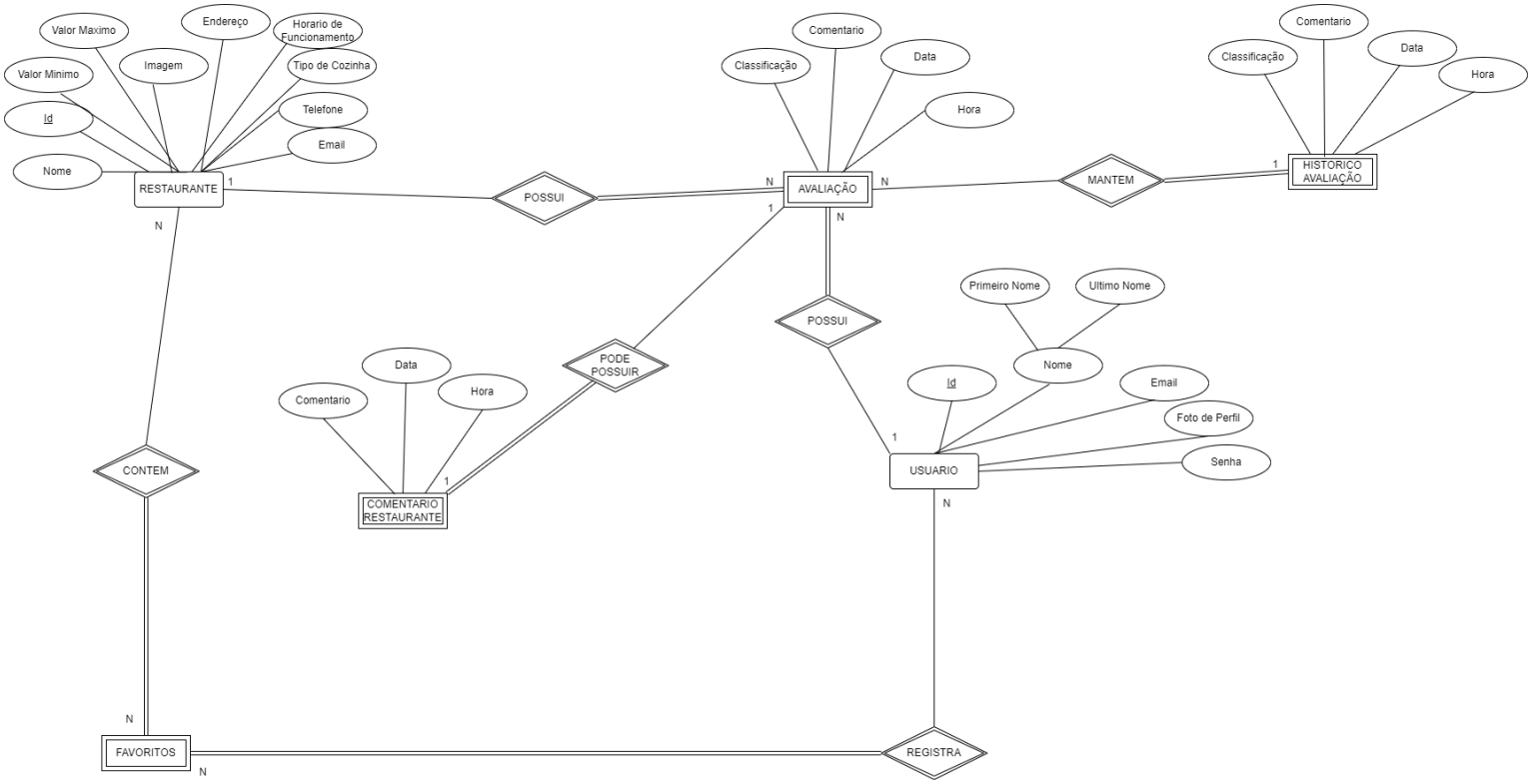
Universo de Discurso do Banco:

O BD-Restaurante proporciona aos usuários um ambiente virtual onde podem realizar as seguintes ações:

1. Um restaurante pode ser avaliado por vários usuários, e um usuário pode avaliar vários restaurantes.
2. Uma avaliação deve incluir uma classificação numérica (por exemplo, de 1 a 5 estrelas) e um comentário escrito.
3. A média das classificações de um restaurante deve ser calculada automaticamente e atualizada sempre que uma nova avaliação for inserida.
4. Cada usuário pode ter apenas uma avaliação para o mesmo restaurante.
5. Os usuários podem ver estatísticas e informações detalhadas sobre restaurantes, incluindo média de classificação, número de avaliações, horários de funcionamento, informações de contato e muito mais.
6. Os usuários podem adicionar restaurantes a uma lista de "Favoritos".
7. O sistema mantém um histórico de todas as avaliações feitas por usuários. Quando um usuário faz uma nova avaliação do mesmo restaurante, a avaliação anterior é adicionada ao histórico.
8. Os usuários devem se registrar para acessar o site.
9. As avaliações e comentários devem incluir data e hora.
10. A pesquisa de restaurantes deve permitir filtrar por nome, faixa de preço e média de avaliações.

O BD-Restaurante visa fornecer uma experiência de usuário agradável e informativa, permitindo que os amantes da comida descubram, compartilhem e explorem o mundo da culinária, promovendo a transparência e a comunicação entre os usuários e os restaurantes. Este banco de dados é projetado para ser uma ferramenta essencial para quem deseja desfrutar de refeições excepcionais e tomar decisões bem informadas ao escolher onde comer.

Diagramação do banco em MER (Modelo Conceitual)



Construção do Banco de Dados Relacional

Tabela Usuario

```
CREATE TABLE IF NOT EXISTS public."user"
(
    id integer NOT NULL DEFAULT nextval('user_id_seq'::regclass),
    first_name text COLLATE pg_catalog."default" NOT NULL,
    last_name text COLLATE pg_catalog."default" NOT NULL,
    profile_picture text COLLATE pg_catalog."default",
    email text COLLATE pg_catalog."default" NOT NULL,
    password text COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "user_PK" PRIMARY KEY (id),
    CONSTRAINT "password_UN" UNIQUE (password)
)
```

Tabela Restaurante

```
CREATE TABLE IF NOT EXISTS public.restaurant
(
    id integer NOT NULL DEFAULT nextval('restaurant_id_seq'::regclass),
    name text COLLATE pg_catalog."default" NOT NULL,
    type_of_kitchen text COLLATE pg_catalog."default" NOT NULL,
    phone text COLLATE pg_catalog."default" NOT NULL,
    email text COLLATE pg_catalog."default" NOT NULL,
    opening_hours text COLLATE pg_catalog."default" NOT NULL,
    address text COLLATE pg_catalog."default" NOT NULL,
    average_rating numeric(2,1),
    image text COLLATE pg_catalog."default",
    min_price numeric(8,2) NOT NULL,
    max_price numeric(8,2) NOT NULL,
    description text COLLATE pg_catalog."default" NOT NULL DEFAULT 'This restaurant
has no description'::text,
    CONSTRAINT "restaurant_PK" PRIMARY KEY (id)
)
```

Tabela Favoritos

```
CREATE TABLE IF NOT EXISTS public.favorites
(
    user_id integer NOT NULL,
    restaurant_id integer NOT NULL,
    CONSTRAINT "favorites_id_PK" PRIMARY KEY (user_id, restaurant_id),
    CONSTRAINT "restaurant_id_FK" FOREIGN KEY (restaurant_id)
        REFERENCES public.restaurant (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "user_id_FK" FOREIGN KEY (user_id)
```

```
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

Tabela Avaliação

```
CREATE TABLE IF NOT EXISTS public.valuation
(
    user_id integer NOT NULL,
    restaurant_id integer NOT NULL,
    rating numeric(2,1) NOT NULL,
    comment text COLLATE pg_catalog."default" NOT NULL,
    date date NOT NULL,
    hour time(4) with time zone NOT NULL,
    CONSTRAINT "valuation_id_PK" PRIMARY KEY (user_id, restaurant_id),
    CONSTRAINT "restaurant_id_FK" FOREIGN KEY (restaurant_id)
        REFERENCES public.restaurant (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "user_id_FK" FOREIGN KEY (user_id)
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT check_rating_range CHECK (rating >= 0.0 AND rating <= 5.0)
)
```

Gatilho registrar avaliação

```
CREATE OR REPLACE TRIGGER trigger_register_valuation
    AFTER INSERT
    ON public.valuation
    FOR EACH ROW
    EXECUTE FUNCTION public.register_valuation();
```

Função registrar avaliação

```
CREATE OR REPLACE FUNCTION public.register_valuation()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
      BEGIN
            INSERT INTO valuation_history (user_id, restaurant_id, rating,
comment, date, hour)
            VALUES (NEW.user_id, NEW.restaurant_id, NEW.rating, NEW.comment,
NEW.date, NEW.hour);
            RETURN NEW;
      END;
$BODY$;
```

Gatilho atualizar média de avaliações do restaurante

```
CREATE OR REPLACE TRIGGER update_restaurant_average_rating_after_insert
    AFTER INSERT
    ON public.valuation
    FOR EACH ROW
    EXECUTE FUNCTION public.update_restaurant_average_rating();
```

Função atualizar média de avaliações do restaurante

```
CREATE OR REPLACE FUNCTION public.update_restaurant_average_rating()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN
  UPDATE restaurant
  SET average_rating = (
    SELECT AVG(rating)
    FROM valuation
    WHERE "restaurant_id" = NEW."restaurant_id"
  )
  WHERE "id" = NEW."restaurant_id";
  RETURN NEW;
END;
$BODY$;
```

Tabela histórico de avaliações

```
CREATE TABLE IF NOT EXISTS public.valuation_history
(
    user_id integer NOT NULL,
    restaurant_id integer NOT NULL,
    rating numeric(2,1) NOT NULL,
    comment text COLLATE pg_catalog."default" NOT NULL,
    date date NOT NULL,
    hour time(4) with time zone NOT NULL,
    CONSTRAINT "valuation_history_id_PK" PRIMARY KEY (user_id, restaurant_id),
    CONSTRAINT "restaurant_id_FK" FOREIGN KEY (restaurant_id)
        REFERENCES public.restaurant (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "user_id_FK" FOREIGN KEY (user_id)
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT check_rating_range CHECK (rating >= 0.0 AND rating <= 5.0) NOT
VALID
)
```

Tabela comentario restaurante

```
CREATE TABLE IF NOT EXISTS public.restaurant_comment
(
    user_id integer NOT NULL,
    restaurant_id integer NOT NULL,
    comment text COLLATE pg_catalog."default" NOT NULL,
    date date NOT NULL,
    hour time(4) with time zone NOT NULL,
    CONSTRAINT "restaurant_comment_PK" PRIMARY KEY (user_id, restaurant_id),
    CONSTRAINT "restaurant_id_FK" FOREIGN KEY (restaurant_id)
        REFERENCES public.restaurant (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "user_id_FK" FOREIGN KEY (user_id)
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

Registrar Histórico de avaliações

```
-- FUNCTION: public.register_valuation()
CREATE OR REPLACE FUNCTION public.register_valuation()
```

```
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
  BEGIN
      INSERT INTO valuation_history (user_id, restaurant_id, rating, comment, date,
hour)
      VALUES (NEW.user_id, NEW.restaurant_id, NEW.rating, NEW.comment, NEW.date,
NEW.hour);
      RETURN NEW;
  END;
$BODY$;

ALTER FUNCTION public.register_valuation()
    OWNER TO postgres;
```

Essa função chamada public.register_valuation() que é uma função automática que ajuda quando você adiciona uma nova avaliação em um restaurante. Aqui está o que ela faz:

Quando ela é acionada:
A função é chamada automaticamente depois que você adiciona uma nova avaliação na tabela de avaliação (valuation).
O que ela faz:
Ela pega as informações dessa nova avaliação (como quem fez a avaliação, a nota, o comentário, etc.) e coloca essas informações em outra tabela chamada valuation history. Essa tabela é como um histórico que guarda todas as avaliações antigas.
Por que isso é útil:
Isso é útil porque mantém um registro de todas as avaliações antigas em um lugar separado. Então, se você quiser ver como um restaurante foi avaliado no passado, pode olhar na tabela valuation_history.

Resumindo, sempre que alguém faz uma avaliação, a função pega essa informação e a guarda em um histórico para referência futura.

Calcular a média de avaliação do restaurante -

```
CREATE OR REPLACE FUNCTION public.update_restaurant_average_rating()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
```

```
   VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN
 UPDATE restaurant
 SET average_rating = (
   SELECT AVG(rating)
   FROM valuation
   WHERE "restaurant_id" = NEW."restaurant_id"
 )
 WHERE "id" = NEW."restaurant_id";
 RETURN NEW;
END;
$BODY$;

ALTER FUNCTION public.update_restaurant_average_rating()
   OWNER TO postgres;
```

Tipo de Gatilho:
Esta função é uma espécie de "ajudante automático" que é chamado sempre que uma nova avaliação é adicionada. O tipo de gatilho associado a essa função geralmente seria AFTER INSERT na tabela valuation.
O que ela faz:
Quando uma nova avaliação é adicionada (NEW), a função pega o restaurant_id dessa avaliação e calcula a nova média das avaliações para aquele restaurante. Então, ela atualiza o campo average_rating na tabela restaurant com essa nova média.
Por que isso é útil:
Manter a média das avaliações diretamente na tabela restaurant é útil porque você pode rapidamente acessar a pontuação média de um restaurante sem ter que calcular toda vez que consulta. Isso pode ser importante para mostrar aos usuários a avaliação média do restaurante.

Então, resumindo, esta função de gatilho atualiza automaticamente a média das avaliações de um restaurante sempre que uma nova avaliação é adicionada, o que pode ser útil para exibir informações precisas de avaliação aos usuários.

O usuário não pode ser deletado enquanto ele ainda tem avaliações-

```
--FUNCTION DONT LET USER BE DELETED
CREATE OR REPLACE FUNCTION prevent_user_deletion()
 RETURNS TRIGGER AS
$BODY$
BEGIN
```

```
 IF EXISTS (SELECT 1 FROM public.valuation V natural join public.user U
         WHERE V.user_id = U.id) THEN
   RAISE EXCEPTION 'Cannot delete user with existing valuations.';
 END IF;


 RETURN OLD;
END;
$BODY$
LANGUAGE plpgsql;


CREATE OR REPLACE TRIGGER prevent_user_deletion_trigger
BEFORE DELETE
ON public.user
FOR EACH ROW
EXECUTE FUNCTION prevent_user_deletion();
```

Tipo de Gatilho:

Esta função é um gatilho do tipo BEFORE DELETE. Isso significa que ela é acionada automaticamente antes que um registro seja excluído da tabela public.user.

O que a Função Faz:

A função prevent_user_deletion verifica se existem avaliações associadas a um usuário antes de permitir sua exclusão.Utiliza um bloco BEGIN e END para agrupar as instruções dentro da função.

A instrução EXISTS verifica se há pelo menos uma linha na tabela public.valuation onde o user_id é igual ao id do usuário que está prestes a ser excluído (OLD.id).

Se existir alguma avaliação associada, a função lança uma exceção usando RAISE EXCEPTION, indicando que não é possível excluir um usuário com avaliações existentes.

Se não houver avaliações associadas, a função retorna o registro OLD, permitindo a continuação do processo de exclusão.

Por que é Útil:

Este gatilho é útil para garantir a integridade referencial. Impede que usuários sejam excluídos quando há avaliações vinculadas a eles, evitando dados inconsistentes.

Ajuda a manter a consistência no banco de dados, garantindo que não haja avaliações "órfãs" sem um usuário associado.

Em resumo, este gatilho e função trabalham juntos para garantir que usuários não sejam excluídos se houverem avaliações associadas, mantendo a integridade dos dados no banco de dados

Não se pode ter restaurantes com nomes iguais-

```
--FUNCTION -- NO RESTAURANTS WITH THE SAME NAME
CREATE OR REPLACE FUNCTION prevent_restaurant_equal_name()
 RETURNS TRIGGER AS
$BODY$
BEGIN

 IF (SELECT 1 FROM public.restaurant R
         WHERE R.name = R.name) THEN
   RAISE EXCEPTION 'Cannot exist restaurants with the same name';
 END IF;


 RETURN OLD;
END;
$BODY$
LANGUAGE plpgsql;


CREATE OR REPLACE TRIGGER prevent_restaurant_equal_name_trigger
BEFORE INSERT
ON public.restaurant
FOR EACH ROW
EXECUTE FUNCTION prevent_restaurant_equal_name();
```

Tipo de Gatilho:

A função é um gatilho do tipo BEFORE INSERT. Isso significa que é acionada automaticamente antes que um novo registro seja inserido na tabela public.restaurant.

O que a Função Faz:

- A função prevent_restaurant_equal_name verifica se já existe algum restaurante na tabela public.restaurant com o mesmo nome antes de permitir a inserção.
- Usa um bloco BEGIN e END para delimitar as instruções dentro da função.

- A instrução IF verifica se existe pelo menos uma linha na tabela public.restaurant onde o nome (R.name) é igual ao nome do restaurante que está prestes a ser inserido (NEW.name).
- Se existir um restaurante com o mesmo nome, a função lança uma exceção usando RAISE EXCEPTION, indicando que não podem existir restaurantes com o mesmo nome.
- Se não existir nenhum restaurante com o mesmo nome, a função retorna o registro OLD, permitindo a continuação do processo de inserção.

Por que é Útil:

- Essa função é útil para garantir que não haja restaurantes com nomes duplicados na tabela. Isso ajuda a manter a integridade dos dados e evita confusões ao buscar ou referenciar restaurantes pelo nome.

Em resumo, esse gatilho e função trabalham juntos para garantir que não existam restaurantes com o mesmo nome na tabela, mantendo a consistência e evitando ambiguidades nos dados do restaurante.

Quando deletar um restaurante tudo relacionado a ele deve ser retirado

```
--FUNCTION -- WHEN DELETED RESTAURANT CLEAR ALL THE IS CLOSE TO HIM
CREATE OR REPLACE FUNCTION delete_restaurant()
 RETURNS TRIGGER AS
$BODY$
BEGIN

  DELETE FROM public.valuation V
         WHERE V.restaurant_id = OLD.id;

  DELETE FROM public.valuation_history VH
        WHERE VH.restaurant_id = OLD.id;

  DELETE FROM public.favorites F
        WHERE F.restaurant_id = OLD.id;




 RETURN OLD;
END;
$BODY$
LANGUAGE plpgsql;


CREATE OR REPLACE TRIGGER clean_traces_deleted_restaurant_trigger
BEFORE DELETE
```

```
ON public.restaurant
FOR EACH ROW
EXECUTE FUNCTION delete_restaurant();
```

Tipo de Gatilho:

A função é um gatilho do tipo BEFORE DELETE. Isso significa que é acionada automaticamente antes que um restaurante seja excluído da tabela public.restaurant.

O que a Função Faz:

- A função delete_restaurant remove todas as informações relacionadas a um restaurante que está prestes a ser excluído.
- Utiliza um bloco BEGIN e END para agrupar as instruções dentro da função.
- Realiza uma sequência de instruções DELETE:
- Remove todas as avaliações (public.valuation) associadas ao restaurante (WHERE V.restaurant_id = OLD.id).
- Remove todas as entradas de histórico de avaliações (public.valuation_history) associadas ao restaurante (WHERE VH.restaurant_id = OLD.id).
- Remove todas as entradas da tabela de favoritos (public.favorites) associadas ao restaurante (WHERE F.restaurant_id = OLD.id).
- A função retorna o registro OLD, permitindo a continuação do processo de exclusão.

Por que é Útil:

- Essa função é útil para garantir uma limpeza completa de todas as informações associadas a um restaurante antes de sua exclusão.
- Evita que informações desnecessárias ou "órfãs" permaneçam no banco de dados após a exclusão do restaurante.

Em resumo, essa função e gatilho trabalham juntos para garantir uma exclusão completa e consistente de um restaurante e suas informações associadas do banco de dados.

Povoamento do Banco

```
INSERT INTO public."user" (first_name, last_name, profile_picture, email,
password)
VALUES
  ('João', 'Sfredo', 'joao.jpg', 'joao@email.com', 'hashed_password_1'),
  ('Lara', 'Rossetti', 'lara.jpg', 'lara@email.com', 'hashed_password_2'),
  ('Gabriel', 'Talim', 'gabriel.jpg', 'gabriel@email.com', 'hashed_password_3'),
  ('Paulo', 'Renato', 'paulo.jpg', 'paulo@email.com', 'hashed_password_4'),
```

```
  ('Matheus', 'Dias', 'matheus.jpg', 'matheus@email.com', 'hashed_password_5');
```

```
INSERT INTO public.restaurant (name, type_of_kitchen, phone, email, opening_hours,
address, image, min_price, max_price, description)
('Restaurante Brasileiro', 'Brasileira', '123456789',
'restauranteBrasileiro@email.com', '08:00 - 22:00', 'Rua A, 123',
'https://imgs.search.brave.com/ym_RnP-a62QM6D3UoLUt9_8O_jqs15oRLqm_Vlk-Ji4/rs:fit:
860:0:0/g:ce/aHR0cHM6Ly9pbWFn/ZXMuc3F1YXJlc3Bh/Y2UtY2RuLmNvbS9j/b250ZW50L3YxLzUx/N
2U5MzM1ZTRiMDg0/NzgyMzUwMDg0NS8x/Mzc5NDIyMDQwMjc3/LVU4VFRQUkdIRUZH/MVhMRDlGOVRBL2N
v/bWlkYS1icmFzaWxl/aXJhLmpwZw',
20.00, 50.00, 'Experimente a autêntica culinária brasileira no nosso restaurante,
onde cada prato é uma viagem pelos sabores exuberantes do Brasil. De feijoadas a
churrascos suculentos, oferecemos uma variedade de pratos que destacam a
diversidade da gastronomia brasileira.'),

('Restaurante Italiano', 'Italiana', '987654321', 'restauranteItaliano@email.com',
'09:00 - 23:00', 'Avenida B, 456',
'https://imgs.search.brave.com/qbhgeK2cXenXBlSt1wHR64_bMOYEilZXPhfLGETSKpU/rs:fit:
860:0:0/g:ce/aHR0cHM6Ly9ibG9n/LmR1b2dvdXJtZXQu/Y29tLmJyL3dwLWNv/bnRlbnQvdXBsb2Fk/c
y8yMDE5LzA3LzM4/LUR1by1Hb3VybWV0/LW1hY2FycmFvLTY3/NXg2NzUuanBn',
15.00, 40.00, 'Bem-vindo ao nosso pedaço da Itália! Descubra a verdadeira paixão
pela comida italiana, onde cada prato é preparado com ingredientes frescos e
sazonais. De massas artesanais a pizzas deliciosas, oferecemos uma autêntica
experiência culinária italiana.'),

('Restaurante Japones', 'Japonesa', '456123789', 'restauranteJapones@email.com',
'11:00 - 20:00', 'Praça C, 789',
'https://imgs.search.brave.com/DrlcK-i499QSs-dP8zRO9Gtfd270w9MPm6Wfa81Aks4/rs:fit:
860:0:0/g:ce/aHR0cHM6Ly9saDUu/Z29vZ2xldXNlcmNv/bnRlbnQuY29tL3Av/QUYxUWlwTWZLWXA4/U
0QwM3ItUGx4OGJP/ZXk1VnZKbURKZk10/T1c5b3pFUzE',
25.00, 60.00, 'Desfrute da elegância e da sofisticação da culinária japonesa em
nosso restaurante. De sushis frescos a pratos tradicionais como ramen e tempura,
nossa cozinha oferece uma experiência única que combina técnica requintada e
ingredientes de alta qualidade.'),

('Restaurante Mexicano', 'Mexicana', '789321456', 'restauranteMexicano@email.com',
'10:00 - 21:00', 'Travessa D, 101',
'https://imgs.search.brave.com/Eui5OdF7COo4CDDEr7KwVEw6r6xR7WQYX19AKaTczO4/rs:fit:
860:0:0/g:ce/aHR0cHM6Ly9pMC53/cC5jb20vZG9ub21l/LmNvbS5ici93cC1j/b250ZW50L3VwbG9h/Z
HMvMjAxMy8wNi9D/T01JREEtMjUyNTIw/TUVYSUNBTkEuanBn/P2ZpdD01MTIsMzQx/JnNzbD0x',
18.00, 45.00, 'Bem-vindo ao nosso vibrante restaurante mexicano, onde cada prato é
uma explosão de sabores e cores. Da guacamole fresca aos tacos picantes, nossa
cozinha mexicana autêntica oferece uma experiência culinária emocionante. Descubra
a paixão e o calor do México em cada mordida.'),
```

```
('Restaurante Indiano', 'Indiana', '321456987', 'restauranteIndiano@email.com',
'12:00 - 22:30', 'Alameda E, 202',
'https://imgs.search.brave.com/5dKBX6HhQccANgQ1zqjm68d-Xn61kjYd3DDOx3VMgj0/rs:fit:
860:0:0/g:ce/aHR0cHM6Ly9tZWRp/YS50aW1lb3V0LmNv/bS9pbWFnZXMvMTAz/NjI5NzMzLzYzMC80/N
zIvaW1hZ2UuanBn',
30.00, 70.00, 'Explore os aromas e os temperos exóticos da Índia em nosso
restaurante indiano. Da rica variedade de curries aos pratos tandoori assados no
forno, oferecemos uma jornada gastronômica que desperta os sentidos. Descubra a
magia da culinária indiana, onde cada prato conta uma história de tradição e
sabor.'),

('Restaurante Chines', 'Chinesa', '654789321', 'restauranteChines@email.com',
'07:00 - 20:00', 'Rua F, 303',
'https://imgs.search.brave.com/gLtMagnujrZ7RrOt_QYUXSLzFWk8nmb0NVwqUVMczf4/rs:fit:
860:0:0/g:ce/aHR0cHM6Ly92ZWph/c3AuYWJyaWwuY29t/LmJyL3dwLWNvbnRl/bnQvdXBsb2Fkcy8y/M
DIxLzEyL1dvbnRv/bi1mcml0by1kZS1m/cmFuZ28tZS1wb3Jj/by1hby1tb2xoby1k/ZS10YW1hcmluZG9
f/My5qcGcuanBnP3F1/YWxpdHk9NzAmc3Ry/aXA9aW5mbyZ3PTY4/MCZoPTQ1MyZjcm9w/PTE',
12.00, 35.00, 'Explore os sabores vibrantes da China em nosso restaurante. De
pratos cantoneses a especialidades picantes de Sichuan, nossa cozinha oferece uma
ampla gama de opções para satisfazer seu paladar. Experimente a diversidade e a
riqueza da culinária chinesa conosco.'),

('Restaurante Frances', 'Francesa', '987456321', 'restauranteFrances@email.com',
'10:30 - 23:00', 'Avenida G, 404',
'https://imgs.search.brave.com/6BdSTFJfZGY8M2lREmWWsRol-a9JeHGJEv8jamNRiNk/rs:fit:
860:0:0/g:ce/aHR0cHM6Ly92aWFn/ZW1lZ2FzdHJvbm9t/aWEuY25uYnJhc2ls/LmNvbS5ici93cC1j/b
250ZW50L3VwbG9h/ZHMvc2l0ZXMvNS8y/MDE3LzA0L2NvbmZp/dC1kZS1jYW5hcmRf/ZnJlZGR5LmpwZw'
,
28.00, 55.00, 'Bem-vindo a um pedaço da França em nosso restaurante. Descubra a
sofisticação e a elegância da cozinha francesa, onde cada prato é uma obra-prima
artesanal. De clássicos como coq au vin a sobremesas irresistíveis, oferecemos uma
experiência gastronômica refinada.'),

('Restaurante Mediterrâneo', 'Mediterrânea', '456987321',
'restauranteMediterrâneo@email.com', '09:00 - 21:30', 'Praça H, 505',
'https://imgs.search.brave.com/UGKwyyD7VG8WujSa3Hs2fP_Bd4L2VCxVQ7K5ALiZ0Gc/rs:fit:
860:0:0/g:ce/aHR0cHM6Ly9yZXMu/Y2xvdWRpbmFyeS5j/b20vdGYtbGFiL2lt/YWdlL3VwbG9hZC93/XzEyMDAsaF82NzQs/Y19maWxsLHFfYXV0/byxmX2F1dG8vcmVz/dGF1cmFudC82ZmQ2/OGQxYi01ZWM0LTQ
4/OWUtYTdlYS0yNjZl/YjQ0M2I4ZmEvOTFl/NmNhMTgtMjIxYi00/NDNiLWFmMjktMjA0/NDc2MzE1MmFm
Lmpw/Zw',
17.00, 42.00, 'Transporte seus sentidos para as margens do Mediterrâneo em nosso
restaurante. Descubra a simplicidade e os sabores frescos da culinária
mediterrânea, com pratos que celebram ingredientes locais, azeites de oliva
aromáticos e ervas frescas.'),

('Restaurante Vegetariano', 'Vegetariana', '123789456',
```

```
'restauranteVegetariano@email.com', '08:30 - 20:30', 'Travessa I, 606',
'https://imgs.search.brave.com/EnDXeoOrKZ0fPlk5zQYMYM5N_RikEYsAW9OfLckudGY/rs:fit:
860:0:0/g:ce/aHR0cHM6Ly9ibG9n/LmR1b2dvdXJtZXQu/Y29tLmJyL3dwLWNv/bnRlbnQvdXBsb2Fk/c
y8yMDE5LzA3LzEx/LUR1by1Hb3VybWV0/LUN1bGluYXJpYS0x/MjAweDY3NS5qcGc',
22.00, 58.00, 'Sabor e saúde se encontram em nosso restaurante vegetariano, onde
cada prato é uma celebração de ingredientes frescos e naturais. De saladas
coloridas a pratos quentes e reconfortantes, oferecemos opções deliciosas para
todos os amantes da culinária vegetariana.'),

('Restaurante Sushi', 'Sushi', '789456123', 'restauranteSushi@email.com', '11:30 -
22:00', 'Alameda J, 707',
'https://imgs.search.brave.com/Nq1wEaotuzSva90ya1AwHwEwbXqMWCMDWLF44N7rdVw/rs:fit:
860:0:0/g:ce/aHR0cHM6Ly9ibG9n/LmdldGluYXBwLmNv/bS5ici93cC1jb250/ZW50L3VwbG9hZHMv/M
jAyMi8xMS8xODIx/MDgzZC1hOWMyLTRj/YjYtODhjOS01N2Nl/MTBmMjBhYWMud2Vi/cA',
32.00, 75.00, 'Entre em um mundo de frescura e criatividade em nosso restaurante
de sushi. Nossos chefs habilidosos combinam técnicas tradicionais com inovação,
oferecendo uma variedade de sushis, sashimis e rolls que satisfazem tanto os
paladares clássicos quanto os mais aventureiros.');
```

Sistema de Informação

O sistema desenvolvido está disponível em:
https://github.com/PaulitoRenatito/Restaurant_Review_System.git
O vídeo demonstrando-o está disponível em:
https://youtu.be/V3C88oTPqPI