OneBitFood V2 – Aula 1

{} onebitcode.com/onebitfood-v2-aula-1

0 – Links importantes

- Guia para instalar as dependências: https://onebitcode.com/guia-de-instalacao-do-ruby-on-rails
- API completa no GitHub: https://github.com/OneBitCodeBlog/onebitfoodV2-api

1 – Planejamento

A ideia inicial

Criar um APP inspirado no Ifood usando Ruby On Rails como API e Next.js como cliente Web

O Planejamento

1. Mockups
    1. Ferramenta: https://www.figma.com
    2. Material: https://drive.google.com/drive/folders/1mqVAQP-96Xa4I8AcdsuFc0v_XJ_elqVx?usp=sharing
2. Modelo do banco de dados
    1. Ferramenta: DbDesigner
    2. Material: https://dbdesigner.page.link/RvxXdi581QjxiDqe8
3. Documentação dos endpoints
    1. Ferramenta: https://www.postman.com
    2. Material: https://documenter.getpostman.com/view/10378249/TzRPk9yD

Dependências

A seguir, veja as dependências para este projeto

- Ruby 3.0.1
- Ruby on Rails 6.0
- Sqlite

Instalação

Caso não tenha as ferramentas instaladas acesse o guia de instalação

2 – Criando o Projeto

Criando o projeto

Nesta etapa você criará e realizará as configurações iniciais do projeto.

1. Crie o projeto executando o seguinte comando

   Default

   ```
   rails new OneBitFood --api
   ```

2. Entre na pasta do projeto

   Default

   ```
   cd OneBitFood
   ```

3. Adicione as seguintes gems ao seu Gemfile

   Default

   ```
   gem 'ransack'
   gem 'rack-cors'
   ```

   - Ransack

     Para realizar as pesquisas

   - Rack Cors

     Para conseguirmos chamar nossa API com segurança através do frontend

4. Instale as gems

   Default

   ```
   bundle install
   ```

Banco de dados

1. Crie o banco de dados executando o comando

Default

```
rails db:create
```

3 – Gerando os Models

Gerando os Models

Nesta etapa você irá gerar os modelos do projeto.

1. Crie um model para **Categoria dos Restaurantes**

Default

```
rails g model Category title
```

2. Crie o model de **Restaurante**

Default

```
rails g model Restaurant name description:text delivery_tax:float city street
neighborhood number complement category:references
```

3. Para que os restaurantes possam gerenciar suas categorias de produtos, crie o model **ProductCategory**

Default

```
rails g model ProductCategory title restaurant:references
```

4. Crie o model **Product** que representará os produtos oferecidos por um restaurante

Default

```
rails g model Product name description:text price:float product_category:references
```

5. Crie o model de **Pedidos**

Default

```
rails g model Order name phone_number total_value:float status:integer restaurant:references city street neighborhood number complement
```

6. Crie um model para gerenciar os **Itens de um Pedido**

Default

```
rails g model OrderProduct quantity:integer order:references product:references
```

Migration

1. Vá até a Migration de criação da tabela Order e adicione um valor padrão para o status de um pedido

   Default

   ```
   t.integer :status, default: 0
   ```

2. Execute as migrations

   Default

   ```
   rails db:migrate
   ```

4 – Relacionamentos das tabelas

Nesta aula você irá configurar os relacionamentos entre os models do projeto.

1 – A mesma Categoria pode ser de **N** Restaurantes. Mapeie o relacionamento adicionando ao model **category.rb**

Default

```
has_many :restaurants
```

2 – Um restaurante pode ter **N** Categorias de Produtos, **N** Pedidos e **N** Avaliações Faça o relcionamento estre essas tabelas adicionando ao model **restaurant.rb**

Default

```
has_many :product_categories
has_many :orders
```

3 – Uma Categoria de Produto pode ser de **N** Produtos Mapeie esta relação adicionando ao model **product_category.rb**

Default

```
1  has_many :products
```

4 – Um Pedido pode ter **N** Itens do Pedido Faça este relacionamento adicionando ao model **order.rb**

Default

```
1  has_many :order_products
```

5 – Um Produto pode estar em **N** Itens do pedido Mapeie o relacionamento adicionando ao model **product.rb**

Default

```
1  has_many :order_products
```

5 – Melhorando os Models

Nesta aula você irá configurar as validações e o enum para os seus Models.

Validações

1. Adicione as validações para o model **Retaurantes**

   app/models/restaurant.rb

   Default

   ```
   1  validates :name, :delivery_tax, :city, :neighborhood, :street, :number,
      presence: true
   ```

2. Faça as validações no model de **Categorias de Restaurante**

   app/models/category.rb

   Default

   ```
   1  validates :title, presence: true
   ```

3. Adicione as validações para o model de **Pedidos**

   app/models/order.rb

   Default

   ```
   1  validates :name, :phone_number, :total_value, :city, :neighborhood, :street,
      :number, presence: true
   ```

4. Realize validações no model de **Itens do Pedido**

   app/models/order_product.rb

   Default

   ```
   1  validates :quantity, presence: true
   ```

5. Faça as validações em **Categoria de Produto**

   app/models/product_category.rb

   Default

   ```
   1  validates :title, presence: true
   ```

6. Adicione as validações para **Produto**

   app/models/product.rb

   Default

   ```
   1  validates :name, :price, presence: true
   ```

Enum

1. Um Pedido pode estar em **andamento** ou **entregue** Para fazer este controle vá ao model **Pedido** e adicione o seguinte enum

Default

```
1  enum status: { waiting: 0, delivered: 1 }
```

6 – Gerando os Controllers

Nesta aula você criará os Controllers e Rotas do seu projeto.

Controllers

Abra o terminal, vá até a pasta do projeto e realize as seguintes instruções

1. Crie o controller **Categories** com a action **index**

   Default

   ```
   rails g controller Categories index --skip-routes
   ```

2. Crie o controller **Restaurants** com as actions **index**, **show** e **search**

   Default

   ```
   rails g controller Restaurants index show --skip-routes
   ```

3. Crie o controller Orders com as actions **create** e **show**

   Default

   ```
   rails g controller Orders create show --skip-routes
   ```

4. Crie o controller AvailableCities com a action index:

   Default

   ```
   rails g controller AvailableCities index --skip-routes
   ```

Rotas

1. Vá até o arquivo **config/routes.rb** e adicione as rotas da sua aplicação

   Default

   ```
   scope "/api", defaults: {format: :json} do
    resources :categories, only: [:index]
    resources :restaurants, only: [:index, :show]
    resources :orders, only: [:create, :show]
    resources :available_cities, only: [:index]
   end
   ```

7 – Imagens

Nesta aula você utilizará o **Active Storage** para permitir que Categorias, Restaurantes, e Produtos tenham uma imagem de apresentação.

Active Storage

1. Faça a instalação do **Active Storage** executando o comando

   Default

   ```
   1  rails active_storage:install
   ```

2. Configure para que os arquivos sejam salvos localmente no anbiente de desenvolvimento

   config/environments/development.rb

   Default

   ```
   1  Rails.application.routes.default_url_options[:host] = 'localhost:3000'
   ```

3. Execute a migration que foi criada

   Default

   ```
   1  rails db:migrate
   ```

Models

Para adicionar a possibilidade de um model possuir uma imagem faça:

1. Vá até o model de **Categorias** e inclua o seguinte código

   app/models/category.rb

   Default

   ```
   1  has_one_attached :image
   ```

2. Faça o mesmo processo para o model de **Restaurante**

   app/models/restaurant.rb

   Default

   ```
   1  has_one_attached :image
   ```

3. E por fim, o mesmo para o model de **Produtos**

   Default

   ```
   1  has_one_attached :image
   ```

Preparando dados de testes

Nessa aula você incluira os dados que serão usados para teste

1. Baixe as imagens do seguinte links (https://drive.google.com/file/d/1vKVyrYw8uMvsfU56n7JjftsvDLNSou-M/view?usp=sharing), descompacte a pasta e coloque em `/public`

2. No arquivo **seeds.rb** coloque:

Default

```
1   Product.destroy_all
2   ProductCategory.destroy_all
3   Restaurant.destroy_all
4   Category.destroy_all
5
6   puts 'Criando Categorias'
7
8   path_image = 'public/images/categories/mexican.jpg'
9   c = Category.create(id: 1, title: 'mexicana')
10  c.image.attach(io: File.open(path_image), filename: 'mexican.jpg')
11
12  path_image = 'public/images/categories/italian.jpeg'
13  c = Category.create(id: 2, title: 'italiana')
14  c.image.attach(io: File.open(path_image), filename: 'italian.jpeg')
15
16  path_image = 'public/images/categories/japonese.jpeg'
17  c = Category.create(id: 3, title: 'japonesa')
18  c.image.attach(io: File.open(path_image), filename: 'japonese.jpeg')
19
20  path_image = 'public/images/categories/vegan.jpeg'
21  c = Category.create(id: 4, title: 'vegana')
22  c.image.attach(io: File.open(path_image), filename: 'vegan.jpeg')
23
24  path_image = 'public/images/categories/peruvian.jpg'
25  c = Category.create(id: 5, title: 'peruana')
26  c.image.attach(io: File.open(path_image), filename: 'peruana.jpg')
27
```

```
28
29  puts 'Cadastrando Restaurantes'
30
31  # Mexican Restaurants
32  path_image = 'public/images/restaurants/1.jpeg'
33  r = Restaurant.create!(
34    name: 'Los Sombreros',
35    description: 'Nossa missão tem sido ajudar as pessoas a alcançar seus
    objetivos de saúde e bem-estar. Embora tenhamos mudado ao longo dos
36  anos, nossos valores permaneceram os mesmos.',
37    delivery_tax: 5.50,
38    city: 'São Paulo', street: 'Bela terra',
39    number: '1393', neighborhood: 'Mercês', category_id: 1
40  )
41  r.image.attach(io: File.open(path_image), filename: '1.jpg')
42  pc = ProductCategory.create!(title: 'Pratos Mexicanos', restaurant: r)
43  prod = Product.create!(name: 'Nacho Guacamole', price: 19, description:
    'Tortilhas com Guacamole',  product_category: pc)
44
    prod.image.attach(io: File.open('public/images/products/nachosg.jpg'),
45  filename: 'nachosg.jpg')
46  prod = Product.create!(name: 'Nacho', price: 19, description: 'Tortilhas com
    milho', product_category: pc)
47
    prod.image.attach(io: File.open('public/images/products/nachosg2.jpeg'),
48  filename: 'nachosg2.jpeg')
49
50  # Curitiba
51  path_image = 'public/images/restaurants/1.jpeg'
52  r = Restaurant.create!(
53    name: 'Los Sombreros - CWB',
54    description: 'Nossa missão tem sido ajudar as pessoas a alcançar seus
    objetivos de saúde e bem-estar. Embora tenhamos mudado ao longo dos
55  anos, nossos valores permaneceram os mesmos.',
56    delivery_tax: 5.50,
57
```

```
58    city: 'Curitiba', street: 'Bela terra',
59    number: '1393', neighborhood: 'Mercês', category_id: 1
60  )
61  r.image.attach(io: File.open(path_image), filename: '1.jpg')
62  pc = ProductCategory.create!(title: 'Pratos Mexicanos', restaurant: r)
63  prod = Product.create!(name: 'Nacho Guacamole', price: 19, description:
    'Tortilhas com Guacamole',  product_category: pc)
64
65  prod.image.attach(io: File.open('public/images/products/nachosg.jpg'),
    filename: 'nachosg.jpg')
66  prod = Product.create!(name: 'Nacho', price: 19, description: 'Tortilhas com
    milho', product_category: pc)
67
68  prod.image.attach(io: File.open('public/images/products/nachosg2.jpeg'),
    filename: 'nachosg2.jpeg')
69
70  path_image = 'public/images/restaurants/2.jpeg'
71  r = Restaurant.create!(
72    name: 'Ola Que Tal',
73    description: 'Para alcançar e manter essa distinção em comida e vinho,
    serviço, ambiente e ambiente, o restaurante ganha reputação de primeira
74  classe por gastronomia, hospitalidade graciosa e informada, conforto e
    beleza que atraem clientes novos e repetidos ano após ano.',
75
76    delivery_tax: 5.50,
77    city: 'São Paulo', street: 'Aminta de Barros',
78    number: '659', neighborhood: 'Centro', category_id: 1
79  )
80  r.image.attach(io: File.open(path_image), filename: '2.jpg')
81  pc = ProductCategory.create!(title: 'Pratos Mexicanos', restaurant: r)
82  prod = Product.create!(name: 'Burrito', price: 19, description: 'Tortilhas com
    Guacamole',  product_category: pc)
83
84  prod.image.attach(io: File.open('public/images/products/bt.jpg'), filename:
    'bt.jpg')
85  prod = Product.create!(name: 'Quesadilha', price: 25, description: 'Tortilhas
    de queijo',  product_category: pc)
86
87  prod.image.attach(io: File.open('public/images/products/quesa.jpeg'),
    filename: 'quesa.jpeg')
```

```
88
89  #Curitiba
90  path_image = 'public/images/restaurants/2.jpeg'
91  r = Restaurant.create!(
92    name: 'Ola Que Tal - CWB',
93    description: 'Para alcançar e manter essa distinção em comida e vinho,
    serviço, ambiente e ambiente, o restaurante ganha reputação de primeira
94  classe por gastronomia, hospitalidade graciosa e informada, conforto e
    beleza que atraem clientes novos e repetidos ano após ano.',
95
    delivery_tax: 5.50,
96
    city: 'Curitiba', street: 'Aminta de Barros',
97
    number: '659', neighborhood: 'Centro', category_id: 1
98
  )
99
  r.image.attach(io: File.open(path_image), filename: '2.jpg')
100
  pc = ProductCategory.create!(title: 'Pratos Mexicanos', restaurant: r)
101
  prod = Product.create!(name: 'Burrito', price: 19, description: 'Tortilhas com
102 Guacamole',  product_category: pc)
103 prod.image.attach(io: File.open('public/images/products/bt.jpg'), filename:
    'bt.jpg')
104
  prod = Product.create!(name: 'Quesadilha', price: 25, description: 'Tortilhas
105 de queijo',  product_category: pc)
106 prod.image.attach(io: File.open('public/images/products/quesa.jpeg'),
107 filename: 'quesa.jpeg')
108
109 # Italian Restaurants
110 path_image = 'public/images/restaurants/3.jpeg'
111 r = Restaurant.create!(
112   name: 'Bravo',
113   description: 'Estamos empenhados em usar os melhores ingredientes em
    nossas receitas. Nenhum alimento deixa a nossa cozinha que nós mesmos
114 não comeríamos.',
115   delivery_tax: 3.50,
116   city: 'São Paulo', street: 'Rua via mar',
117   number: '250', neighborhood: 'Centro', category_id: 2
```

```
118 )
119 r.image.attach(io: File.open(path_image), filename: '3.jpg')
120 pc = ProductCategory.create!(title: 'Porções', restaurant: r)
121 prod = Product.create!(name: 'Berinjela à parmegiana', price: 78,
    description: 'Com arroz e fritas',  product_category: pc)
122
123 prod.image.attach(io: File.open('public/images/products/berinjela.jpg'),
    filename: 'berinjela.jpg')
124 prod = Product.create!(name: 'Fritas', price: 35, description: 'Bata frita com
    bacon',  product_category: pc)
125
126 prod.image.attach(io: File.open('public/images/products/fritas.jpg'),
    filename: 'fritas.jpg')
127
128 #Maceio
129 path_image = 'public/images/restaurants/3.jpeg'
130 r = Restaurant.create!(
131   name: 'Bravo - Maceio',
132   description: 'Estamos empenhados em usar os melhores ingredientes em
    nossas receitas. Nenhum alimento deixa a nossa cozinha que nós mesmos
133 não comeríamos.',
134   delivery_tax: 3.50,
135   city: 'Maceio', street: 'Rua via mar',
136   number: '250', neighborhood: 'Centro', category_id: 2
137 )
138 r.image.attach(io: File.open(path_image), filename: '3.jpg')
139 pc = ProductCategory.create!(title: 'Porções', restaurant: r)
140 prod = Product.create!(name: 'Berinjela à parmegiana', price: 78,
141 description: 'Com arroz e fritas',  product_category: pc)
142 prod.image.attach(io: File.open('public/images/products/berinjela.jpg'),
    filename: 'berinjela.jpg')
143 prod = Product.create!(name: 'Fritas', price: 35, description: 'Bata frita com
144 bacon',  product_category: pc)
145 prod.image.attach(io: File.open('public/images/products/fritas.jpg'),
    filename: 'fritas.jpg')
146
147
```

```
path_image = 'public/images/restaurants/4.jpeg'
r = Restaurant.create!(
  name: 'La Pergoletti',
  description: 'Nossa missão é estabelecer relações comerciais benéficas com diversos fornecedores que compartilham nosso compromisso com o atendimento ao cliente, qualidade e preços competitivos.',
  delivery_tax: 6.70,
  city: 'São Paulo', street: 'Rua Joaquim Pinto',
  number: '929', neighborhood: 'Vila Gomes Cardim', category_id: 2
)
r.image.attach(io: File.open(path_image), filename: '4.jpg')
pc = ProductCategory.create!(title: 'Fogazzas (Individuais)', restaurant: r)
prod = Product.create!(name: 'Fogazza Bacon', price: 12, description: 'Bacon, parmesão e mussarela.',  product_category: pc)
prod.image.attach(io: File.open('public/images/products/fogazza.jpg'), filename: 'fogazza.jpg')
prod = Product.create!(name: 'Fogazza A moda da Casa', price: 12, description: 'Calabresa, bacon, palmito e mussarela.',  product_category: pc)
prod.image.attach(io: File.open('public/images/products/fogazza.jpg'), filename: 'fogazza.jpg')

# Japonese Restaurants
path_image = 'public/images/restaurants/5.jpeg'
r = Restaurant.create!(
  name: 'Sushi Eterno',
  description: 'Existimos para garantir que cada hóspede receba um serviço rápido, profissional, amigável e cortês.',
  delivery_tax: 7.50,
  city: 'São Paulo', street: 'Avenida Manoel Domingos Pinto',
  number: '507', neighborhood: 'Parque Anhangüera', category_id: 3
)
r.image.attach(io: File.open(path_image), filename: '5.jpg')

```

```
pc = ProductCategory.create!(title: 'Entrada', restaurant: r)
prod = Product.create!(name: 'Temaki', price: 19.99, description: 'Enrolado de arroz com alga marinha em forma de cone',  product_category: pc)
prod.image.attach(io: File.open('public/images/products/temaki.jpeg'), filename: 'temaki.jpeg')
prod = Product.create!(name: 'Sashimi', price: 30.90, description: 'Peixe cru fatiado, salmao, atum e peixe prego',  product_category: pc)
prod.image.attach(io: File.open('public/images/products/sashimi.jpg'), filename: 'sashimi.jpg')

#Maceio
path_image = 'public/images/restaurants/5.jpeg'
r = Restaurant.create!(
  name: 'Sushi Eterno - Maceio',
  description: 'Existimos para garantir que cada hóspede receba um serviço rápido, profissional, amigável e cortês.',
  delivery_tax: 7.50,
  city: 'Maceio', street: 'Avenida do mar',
  number: '2344', neighborhood: 'Belo mar', category_id: 3
)
r.image.attach(io: File.open(path_image), filename: '5.jpg')
pc = ProductCategory.create!(title: 'Entrada', restaurant: r)
prod = Product.create!(name: 'Temaki', price: 19.99, description: 'Enrolado de arroz com alga marinha em forma de cone',  product_category: pc)
prod.image.attach(io: File.open('public/images/products/temaki.jpeg'), filename: 'temaki.jpeg')
prod = Product.create!(name: 'Sashimi', price: 30.90, description: 'Peixe cru fatiado, salmao, atum e peixe prego',  product_category: pc)
prod.image.attach(io: File.open('public/images/products/sashimi.jpg'), filename: 'sashimi.jpg')

path_image = 'public/images/restaurants/6.jpeg'
r = Restaurant.create!(
  name: 'Okuyamah',
```

```
208   description: 'Restaurante conceituado, vencedor por 5 vezes como melhor
      restaurante Japones de São Paulo.',
209
      delivery_tax: 8.30,
210
      city: 'São Paulo', street: 'Rua Francisco Artassio',
211
      number: '134', neighborhood: 'Jardim das Laranjeiras', category_id: 3
212
    )
213
    r.image.attach(io: File.open(path_image), filename: '6.jpg')
214
    pc = ProductCategory.create!(title: 'Entrada', restaurant: r)
215
    prod = Product.create!(name: 'Hossomaki 16 unidades', price: 20.90,
216 description: 'Enrolado fino com folha de alga marinha por
    fora.',  product_category: pc)
217
    prod.image.attach(io: File.open('public/images/products/hosomaki.jpg'),
218 filename: 'hosomaki.jpg')
219 prod = Product.create!(name: 'Hot roll - 10 unidades', price: 12, description:
    '10 unidades.',  product_category: pc)
220
    prod.image.attach(io: File.open('public/images/products/hot-holl.jpg'),
221 filename: 'hot-holl.jpg')
222
223 #Maceio
224 path_image = 'public/images/restaurants/6.jpeg'
225 r = Restaurant.create!(
226   name: 'Okuyamah - Maceio',
227   description: 'Restaurante conceituado, vencedor por 5 vezes como melhor
    restaurante Japones de São Paulo.',
228
229   delivery_tax: 8.30,
230   city: 'Maceio', street: 'Rua Francisco Artassio',
231   number: '134', neighborhood: 'Jardim das Laranjeiras', category_id: 3
232 )
233 r.image.attach(io: File.open(path_image), filename: '6.jpg')
    pc = ProductCategory.create!(title: 'Entrada', restaurant: r)
    prod = Product.create!(name: 'Hossomaki 16 unidades', price: 20.90,
    description: 'Enrolado fino com folha de alga marinha por
    fora.',  product_category: pc)
```

```
prod.image.attach(io: File.open('public/images/products/hosomaki.jpg'),
filename: 'hosomaki.jpg')

prod = Product.create!(name: 'Hot roll - 10 unidades', price: 12, description:
'10 unidades.',  product_category: pc)

prod.image.attach(io: File.open('public/images/products/hot-holl.jpg'),
filename: 'hot-holl.jpg')


# Vegan Restaurants

path_image = 'public/images/restaurants/7.jpeg'

r = Restaurant.create!(

  name: 'Club Life',

  description: 'NOSSA ESPECIALIDADE. pratos vegetais de alta
qualidade,com opções de alimentos integrais, sem glúten e sem lactose.',

  delivery_tax: 5.70,

  city: 'São Paulo', street: 'Alameda dos Uapês',

  number: '933', neighborhood: 'Planalto Paulista', category_id: 4

)

r.image.attach(io: File.open(path_image), filename: '7.jpg')

pc = ProductCategory.create!(title: 'Saladas, molhos e wraps', restaurant: r)

prod = Product.create!(name: 'Coleslaw', price: 8.99, description: 'Repolho
roxo, couve, cenoura, cebola, maionese de castanha e
xylitol',  product_category: pc)

prod.image.attach(io: File.open('public/images/products/coleslaw.jpg'),
filename: 'coleslaw.jpg')

prod = Product.create!(name: 'Side salad', price: 9.90, description: 'Mix de
folhas com cenoura ralada, tomatinho sweet e semente de
girassol.',  product_category: pc)

prod.image.attach(io: File.open('public/images/products/side-salad.jpeg'),
filename: 'side-salad.jpeg')


path_image = 'public/images/restaurants/8.jpeg'

r = Restaurant.create!(

  name: 'Casa Natural',
```

```
  description: 'Oferecemos, desde 1981, refeições ovo-lacto-vegetarianas, leves, saudáveis, balanceadas e principalmente saborosas, procurando aliar o sabor, a qualidade de vida e o bem-estar dos clientes.',

  delivery_tax: 8.30,

  city: 'São Paulo', street: 'Rua Natal',

  number: '938', neighborhood: 'Cantinho do Céu', category_id: 4

)

r.image.attach(io: File.open(path_image), filename: '8.jpg')

pc = ProductCategory.create!(title: 'Saladas, molhos e wraps', restaurant: r)

prod = Product.create!(name: 'Salada de quinoa', price: 20.90, description: 'Alface americana, roxa, frisee, quinoa cozida, cenoura, tomate, damasco dessecado, amendoa crua.',  product_category: pc)

prod.image.attach(io: File.open('public/images/products/salada-de-quinoa.jpg'), filename: 'salada-de-quinoa.jpg')

prod = Product.create!(name: 'Coleslaw', price: 11, description: 'Repolho roxo, couve, cenoura, cebola, maionese de castanha e xylitol',  product_category: pc)

prod.image.attach(io: File.open('public/images/products/coleslaw2.jpeg'), filename: 'coleslaw2.jpeg')
```

3. Inclua no banco de dados:

   Default

   ```
   1  rails db:seed
   ```

   Obs: Se você tiver algum problema para rodar os seeds de uma única vez, rode **rails c** e rode parte a parte dos seeds.

8 – Preparando os controllers

Nesta aula vamos incluir os métodos para devolver os restaurantes e as categorias

1. Adicione o seguinte código a action index, no controller de **Categorias**

   app/controllers/categories_controller.rb

   Default

```
1  @categories = Category.all.order(:title)
2  render json: @categories
```

2. Crie o arquivo index.json.jbuilder em /views/categories e coloque nele:

   Default

```
1  json.array! @categories do |category|
2   json.id category.id
3   json.title category.title
4   json.image_url polymorphic_url(category.image) if category.image.attached?
5  end
```

3. Adicione o seguinte código a action index, no controller *Restaurants*:

   app/controllers/restaurants_controller.rb

   Default

```
1  @restaurants = Restaurant.all
```

4. Crie o arquivo index.json.jbuilder em /views/restaurantes e coloque nele:

   Default

```
1  json.array! @restaurants do |restaurant|
2   json.partial! restaurant
3  end
```

5. Crie o arquivo _restaurant.json.jbuilder em /views/restaurantes e coloque nele:

Default

```
1  json.id restaurant.id
2  json.name restaurant.name
3  json.description restaurant.description
4  json.status restaurant.status
5  json.delivery_tax restaurant.delivery_tax
6  json.city restaurant.city
7  json.street restaurant.street
8  json.neighborhood restaurant.neighborhood
9  json.number restaurant.number
10 json.complement restaurant.complement
11 json.reference restaurant.reference
12 json.cep restaurant.cep
13
14 json.review restaurant.reviews&.average(:value).round()
15 json.image_url polymorphic_url(restaurant.image) if
   restaurant.image.attached?
16
17
   json.product_categories restaurant.product_categories do
18 |product_category|
19  json.partial! 'product_categories/product_category', product_category:
   product_category

   end
```

6. Crie o arquivo _product_category.json.jbuilder em /views/product_categories e coloque nele:

Default

```
1  json.id product_category.id
2  json.title product_category.title
3
4  json.products product_category.products do |product|
5   json.partial! 'products/product', product: product
6  end
```

7. Crie o arquivo _product.json.jbuilder em /views/products e coloque nele:

Default

```
1  json.id product.id
2  json.name product.name
3  json.description product.description
4  json.price product.price
5  json.image_url polymorphic_url(product.image) if product.image.attached?
```

8. No Controller **app/controllers/restaurants_controller.rb** adicione:

Default

```
1  before_action :set_restaurant, only: :show
```

9. Inclua nos métodos privados do controller:

Default

```
1  def set_restaurant
2   @restaurant = Restaurant.find(params[:id])
3  end
```

10. Crie o arquivo show.json.jbuilder em /views/restaurantes e coloque nele:

    Default

    ```
    1  json.partial! @restaurant
    ```

11. No Controller **app/controllers/available_cities.rb** na action index adicione:

    Default

    ```
    1  @available_cities = Restaurant.all.map { |r| r.city }.uniq
    ```

12. Crie a seguinte estrutura:

    Default

    ```
    1  mkdir views/available_cities
    2  touch views/available_cities/index.json.jbuilder
    ```

13. Coloque no arquivo criado:

    Default

    ```
    1  json.available_cities @available_cities
    ```

9 – Filtros

Nesta aula você criará os filtros por Categorias, Restaurantes e Cidade

1. No método index do controller **Restaurantes** adicione

   app/controllers/restaurants_controller.rb

   Default

```
def index
 @restaurants = Restaurant.all
 filter_by_query if params[:q]
 filter_by_city if params[:city]
 filter_by_category if params[:category]
end
```

2. Agora coloque nos métodos privados:

   Default

```
 def filter_by_query
  @restaurants = @restaurants.ransack(name_or_description_cont:
params[:q]).result
 end


 def filter_by_city
  @restaurants = @restaurants.where(city: params[:city])
 end


 def filter_by_category
  @restaurants = @restaurants.select do |r|
    r.category.title == params[:category]
  end
 end
```

10 – Controller Order

Nessa aula você implementará a funcionalidade de **Realizar um Pedido**

1. Adicione um **callback** para salvar o valor total do pedido.

   Vá até o model de **Pedido** e adicione o seguninte código

   app/models/order.rb

   Default

```
1   before_validation :set_price
2
3   private
4
5   def set_price
6    final_price = 0
7    order_products.each do |order_product|
8      final_price += order_product.quantity * order_product.product.price
9    end
10
11   self.total_value = final_price
12  end
```

2. Para permitir que os **Itens de um Pedido** sejam salvos junto ao **Pedido** utilize **Nested Attributes**

   Vá até o model **order.rb** e adicione a linha a seguir

   Default

```
1  accepts_nested_attributes_for :order_products, allow_destroy: true
```

3. Agora adicione o seguinte código ao controller de **Pedidos**

   Default

```ruby
class OrdersController < ApplicationController
 before_action :set_order, only: :show

 def create
  order = Order.new(order_params)
  if order.save
   render json: @order, status: :created
  else
   render json: order.errors, status: :unprocessable_entity
  end
 end

 def show
  render json: @order
 end

 private

 def set_order
  @order = Order.find(params[:id])
 end

 def order_params
  params.require(:order).permit(
   :name, :phone_number, :restaurant_id, :address,
   order_products_attributes: %i[quantity comment product_id]
  )
 end
end
```

4. Crie o arquivo _show.json.jbuilder em /views/orders e coloque nele:

   Default

   ```
   1  json.partial! @order
   ```

5. Crie o arquivo _order.json.jbuilder em /views/orders e coloque nele:

   Default

   ```
   1  json.id order.id
   2  json.restaurant_id order.restaurant_id
   3  json.total_value order.total_value
   4  json.status order.status
   5  json.address order.address
   ```

11 – Configurando o CORS na API

Nesta aula vamos ajustar a API para que ela permita chamadas vindas do javascript do Browser

2. Instale a gem rodando:

   Default

   ```
   1  bundle install
   ```

3. Agora substitua o conteúdo do arquivo config/initializers/cors.rb por:

Default

```
Rails.application.config.middleware.insert_before 0, Rack::Cors do
 allow do
   origins '*'

   resource '*',
     headers: :any,
     methods: [:get, :post, :put, :patch, :delete, :options, :head]
 end
end
```