



PMR3304 - Sistemas de Informação - 05

Aula 05

Prof. Dr. Marcos de Sales Guerra Tsuzuki

13 de Setembro de 2019

PMR-EPUSP

Desenvolvendo a Relação entre Modelos

Vamos criar um novo projeto

```
rails new students
```

avance para o projeto recém criado: **cd students.**

Crie um modelo

```
rails generate scaffold student given_name:string  
  middle_name:string family_name:string date_of_birth:  
  date grade_point_average:decimal start_date:date
```



Para você fazer em casa

1 Execute os comandos descritos.

Vamos criar um novo modelo

```
rails generate scaffold award name:string year:integer  
  student_id:integer
```



Para você fazer em casa

| Execute os comandos descritos.

Conectando Awards e Students - Introdução

Vamos estabelecer uma relação. Vamos para o arquivo **app/models/student.rb**.

```
class Student < ApplicationRecord
  # a student can have many awards
  has_many :awards
end
```

E no arquivo **app/models/awards.rb**.

```
class Award < ApplicationRecord
  # every award id linked to a student, through student_id
  belongs_to :student
end
```

Vamos executar o migrate: **rails db:migrate**. Em seguida execute o servidor **rails server**. Vamos verificar o que ocorre ao inspecionar a URL

```
http://localhost:3000/students/new.
```

Neste momento, crie um *student* que será utilizado nas atividades posteriores. Verifique agora a criação de um *award* pela inspeção da URL

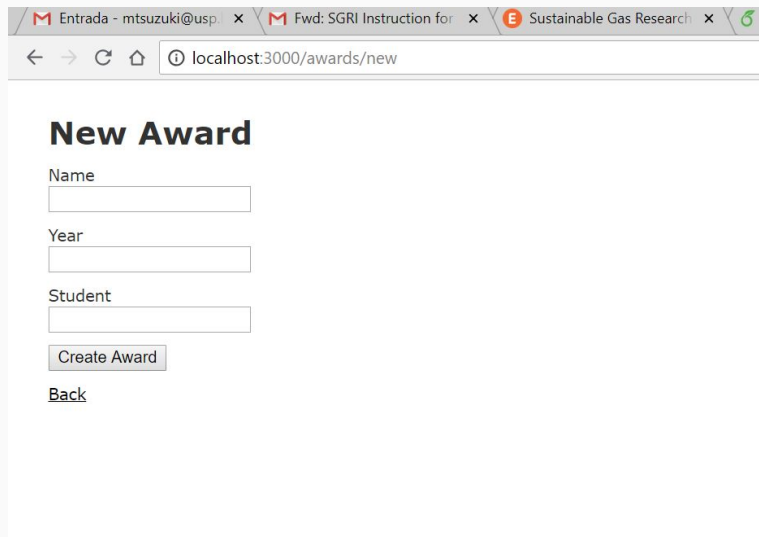
```
http://localhost:3000/awards/new.
```



Para você fazer em casa

- 1 Execute os comandos descritos.

Conectando Awards e Students - Introdução



The screenshot shows a web browser window with three tabs: 'Entrada - mtsuzuki@usp.', 'Fwd: SGRI Instruction for', and 'Sustainable Gas Research'. The address bar shows 'localhost:3000/awards/new'. The page content includes a heading 'New Award', three text input fields labeled 'Name', 'Year', and 'Student', a 'Create Award' button, and a 'Back' link.

Entrada - mtsuzuki@usp. x Fwd: SGRI Instruction for x Sustainable Gas Research x

localhost:3000/awards/new

New Award

Name

Year

Student

Create Award

[Back](#)

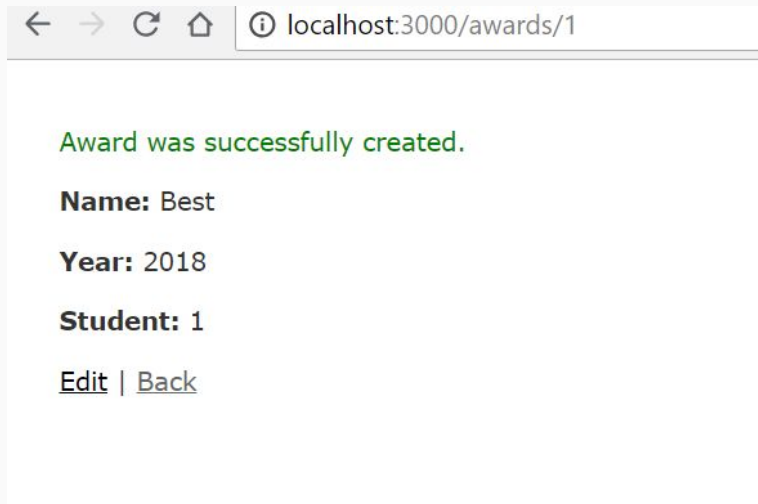
Podemos melhorar a associação de um *student* ao *award*, acrescentando um combo box do tipo *select* no arquivo **app/views/awards/_form.html.erb**.

```
<div class="field">
  <%= form.label :student_id %><br/>
  <%= form.select :student_id, Student.all.order(
    :family_name, :given_name).collect{|s|[
      (s.given_name + " " + s.family_name), s.id]} %>
</div>
```



Para você fazer em casa

I Teste a nova funcionalidade.



Para consertar isto, vamos modificar o arquivo **app/views/awards/show.html.erb**. A linha contendo

```
<%= @award.student_id %>
```

Ficará como

```
<%= @award.student.given_name %> <%= @award.student.  
family_name %>
```

O novo método *student* foi reconhecido porque temos o método **belongs_to** existente em *awards*.

Também será necessário modificar o arquivo **app/views/awards/index.html.erb**. A linha contendo

```
<%= @award.student_id %>
```

Ficará como

```
<%= @award.student.given_name %> <%= @award.student.  
family_name %>
```



Para você fazer em casa

- I Teste a nova funcionalidade.

O código poderá ser simplificado se criarmos um método **name** no arquivo **app/models/student.rb**.

```
def name
  given_name + " " + family_name
end
```

O código nos arquivos **app/views/awards/index.html.erb** e **app/views/awards/show.html.erb** ficará como:

```
<%= @award.student.name %>
```

A criação do combo box em **app/views/awards/_form.html.erb** também ficará mais simples.

```
<%= form.select :student_id, Student.all.collect{|s| [s.name, s.id]} %>
```

O código nos arquivos **app/views/awards/index.html.erb** e **app/views/awards/show.html.erb** ficará como:

```
<%= @award.student.name %>
```



Para você fazer em casa

- 1 Teste a nova funcionalidade.

Quando um *student* é removido, os seus *awards* também devem ser removidos. O Rails possui uma opção simples para realizar este procedimento, é necessário acrescentar uma opção à declaração **has_many** em **app/models/student.rb**.

```
has_many :awards, dependent: :destroy
```

Vamos acrescentar um contador do número de *awards* que um *student* possui. Deverá surgir uma nova coluna à lista de *students*. Assim, devemos modificar o arquivo **app/views/students/index.html.erb**.

```
<th>Start date</th>
```

```
<th>Awards</th>
```

...

```
<td><%= student.start_date %></td>
```

```
<td><%= student.awards.count %></td>
```



Para você fazer em casa

| Crie pelo menos 3 *students* e 2 *awards* para cada *student*.

Vamos melhorar a formatação. É adequado também acrescentar uma lista de *awards* para a **view** cada *student*. Assim, cada usuário poderá observar os *awards* de cada *student*. Vamos modificar o arquivo **app/views/student/show.html.erb**.

```
<h3>Awards</h3><table><tr>
<th>Name</th><th>Year</th><th>Student</th>
<% @student.awards.each do |award| %>
<tr><td><%= award.name %></td><td><%= award.year %></td>
<td><%= award.student.name %></td></tr><% end %>
</table>
```


Vamos observar o arquivo **config/routes.rb**

```
Rails.application.routes.draw do  
  resources :awards  
  resources :students  
end
```

Vamos modificar este arquivo para

```
Rails.application.routes.draw do  
  resources :students do  
    resources :awards  
  end  
end
```

Assim, será possível visitar a URL `http://localhost:3000/students`, mas ao tentar acessar `http://localhost:3000/awards` surgirá um erro.

Entretanto, muitas modificações adicionais serão necessárias no **controller**. Vamos realizar modificações no arquivo **`app/controllers/awards_controller.rb`**.

```
class AwardsController < ApplicationController
  before_action :get_student
  before_action :set_award, only: [:show, :edit, :update, :
    destroy]
  def index
    @awards = @student.awards
    # era @awards = Award.all
  end
  ...
  def new
    @award = @student.awards.build
    # era @award = Award.new
  end
```

Conectando Awards e Students - Modificando o Controller

```
def create
  @award = @student.awards.build(award_params)
  # era @award = Award.new(award_params)
  respond_to do |format|
    if @award.save
      format.html { redirect_to student_awards_url(
        @student), notice: Award was successfully
        created. }
      # era format.html { redirect_to @award, notice: 'Award
        was successfully created.' }
      format.json { render :show, status: :created, location:
        @award }
    else
```

```
def update
  respond_to do |format|
    if @award.update(award_params)
      format.html { redirect_to student_awards_url(
        @student), notice: 'Award was successfully
        updated.' }
      # era format.html { redirect_to @award, notice: 'Award
        was successfully updated.' }
      format.json { render :show, status: :ok, location:
        @award }
    else
```

```
private
  def get_student
    @student = Student.find(params[:student_id])
  end

  # Use callbacks to share common setup or constraints
  between actions.

  def set_award
    @award = @student.awards.find(params[:id])
    # era @award = Award.find(params[:id])
  end
```

As modificações que foram feitas convertem uma referência para *awards* em uma referência para *awards* segundo um *student*. Você verificará que algumas inconsistências de nome para os dois casos. O caso **Award.all** não existirá mais. O método **new** ficará **build**. **awards_url** ficará como **student_awards_url**.

Estes novos nomes aparecem e foram criados automaticamente pelo Rails devido ao novo roteamento.

O novo **AwardsController** utiliza uma nova técnica. Ele inicia com a execução da **before_action**, neste caso acionando o **get_student**. O controller recebe o **student_id** pelo roteamento (pela URL).

Precisamos modificar também a view dos *awards*, que corresponde ao arquivo **app/views/awards/index.html.erb**.

```
<p id="notice"><%= notice %></p>

<h1>Awards for <%= @student.name %> </h1>

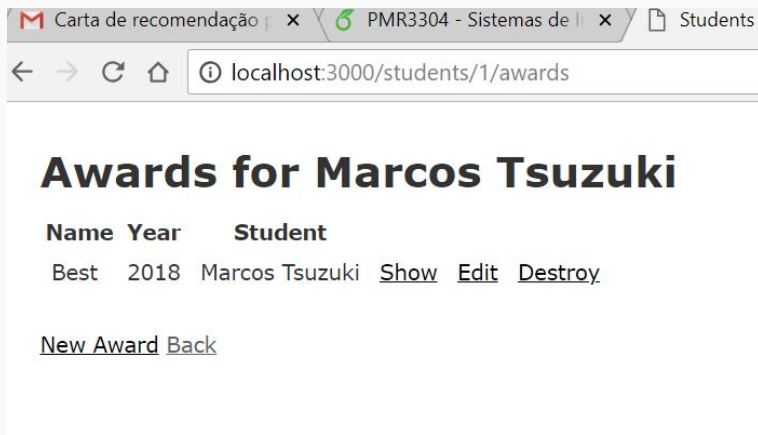
<% if !@student.awards.empty? %>
  <table>
```

Modificar apenas as linhas do meio do código.


```
<td><%= link_to 'Show', [ @student, award ] %></td>  
<td><%= link_to 'Edit', edit_student_award_path( @student,  
  award ) %></td>  
<td><%= link_to 'Destroy', [ @student, award ], method: :  
  delete, data: { confirm: 'Are you sure?' } %></td>
```

```
<br>
<% else %>
  <p><%= @student.given_name %> hasn't won any awards
    yet.</p>
<% end %>
<%= link_to 'New Award', new_student_award_path(@student)
  %>
<%= link_to 'Back', @student %>
```

Conectando Awards e Students - Modificando as Views



The screenshot shows a web browser window with three tabs: 'Carta de recomendação', 'PMR3304 - Sistemas de I', and 'Students'. The address bar shows 'localhost:3000/students/1/awards'. The main content area displays the title 'Awards for Marcos Tsuzuki' in a large, bold font. Below the title is a table with columns 'Name', 'Year', and 'Student'. The table contains one row with the values 'Best', '2018', and 'Marcos Tsuzuki'. To the right of the table are three links: 'Show', 'Edit', and 'Destroy'. Below the table are two more links: 'New Award' and 'Back'.

Name	Year	Student			
Best	2018	Marcos Tsuzuki	Show	Edit	Destroy

[New Award](#) [Back](#)



Para você fazer em casa

Teste em casa o uso da view, inclusive para um *student* sem *awards*.

Precisamos modificar os links do arquivo

app/view/awards/show.html.erb:

```
<%= link_to 'Edit', edit_student_award_path(@student,  
  @award) %> |  
<%= link_to 'Back', student_awards_path(@student) %>
```

Também temos modificações menores no arquivo
app/view/awards/new.html.erb

```
<h1>New Award for <%= @student.name %></h1>  
  
<%= render 'form', award: @award %>  
  
<%= link_to 'Back', student_awards_path(@student) %>
```

Também temos modificações menores no arquivo **app/view/awards/edit.html.erb**

```
<h1>Editing Award for <%= @student.name %></h1>  
  
<%= render 'form', award: @award %>  
  
<%= link_to 'Show', [@student, @award] %> |  
<%= link_to 'Back', student_awards_path(@student) %>
```

Devemos realizar uma pequena modificação no arquivo **app/view/awards/_form.html.erb**

```
<%= form_for([@student, award]) do |form| %>
```

A seleção do **:student_id** também deve ser removida.



Para você fazer em casa

- I Teste em casa o uso da view.

Agora com as modificações, foi criada uma relação estreita entre *students* e *awards*. Entretanto, ainda não estão totalmente integrados. Falta criar uma conexão de *students* para *awards*. A inclusão de novos links faz sentido claro apenas em dois lugares: 1. na listagem e, 2. na view que exhibe cada *student*. Abaixo, temos o código para o arquivo **app/view/students/show.html.erb** (apenas a linha do meio é nova).

```
<%= link_to 'Edit', edit_student_path(@student) %> |  
<%= link_to 'Awards', student_awards_path(@student) %> |  
<%= link_to 'Back', students_path %>
```


Abaixo, temos o código para o arquivo **app/view/students/index.html.erb** (apenas a linha do meio é nova).

```
<td><%= link_to 'Edit', edit_student_path(student) %></td>  
<td><%= link_to 'Awards', student_awards_path(student) %></td>  
<td><%= link_to 'Destroy', student, method: :delete, data: {  
  confirm: 'Are you sure?' } %></td>
```

The End!