

Relatório Sprint-3 CAMAAR

Erick Hideki Taira 222011525

Davi de Araujo Garcez Bueno 211060586

Guilherme de Melo Cardozo 211020885

Joao Pedro Felicio Pereira 202065897

Introdução

A aplicação CAMAAR é um sistema de gerenciamento de questionários para avaliação de cursos e professores. O sistema permite que os coordenadores de curso criem questionários personalizados e os distribuam para os alunos e professores. Os alunos e professores podem responder aos questionários. O sistema também permite que os coordenadores de curso visualizem os resultados dos questionários e façam análises.

Entidades Principais

1. Coordenador

- **id**: Identificador único (integer, chave primária)
- **departamento_id**: Chave estrangeira para Departamento (integer)
- **user_id**: Chave estrangeira para User (integer)
- **created_at**: Data de criação (datetime)
- **updated_at**: Data de atualização (datetime)

2. Departamento

- **id**: Identificador único (integer, chave primária)
- **nome**: Nome do departamento (string)
- **created_at**: Data de criação (datetime)
- **updated_at**: Data de atualização (datetime)

3. Materium

- **id**: Identificador único (integer, chave primária)
- **nome**: Nome da matéria (string)

- **codigo:** Código da matéria (string)
- **departamento_id:** Chave estrangeira para Departamento (integer)
- **created_at:** Data de criação (datetime)
- **updated_at:** Data de atualização (datetime)

4. Matricula

- **id:** Identificador único (integer, chave primária)
- **user_id:** Chave estrangeira para User (integer)
- **turma_id:** Chave estrangeira para Turma (integer)
- **created_at:** Data de criação (datetime)
- **updated_at:** Data de atualização (datetime)

5. Questao

- **id:** Identificador único (integer, chave primária)
- **texto:** Texto da questão (text)
- **enunciado:** Enunciado da questão (text)
- **template_id:** Chave estrangeira para Template (integer)
- **created_at:** Data de criação (datetime)
- **updated_at:** Data de atualização (datetime)

6. Questionario

- **id:** Identificador único (integer, chave primária)
- **nome:** Nome do questionário (string)
- **turma_id:** Chave estrangeira para Turma (integer)
- **template_id:** Chave estrangeira para Template (integer)
- **user_id:** Chave estrangeira para User (integer)
- **created_at:** Data de criação (datetime)
- **updated_at:** Data de atualização (datetime)

7. Respostum

- **id**: Identificador único (integer, chave primária)
- **valor**: Valor da resposta (string)
- **questao_id**: Chave estrangeira para Questao (integer)
- **questionario_id**: Chave estrangeira para Questionario (integer)
- **user_id**: Chave estrangeira para User (integer)
- **created_at**: Data de criação (datetime)
- **updated_at**: Data de atualização (datetime)

8. Template

- **id**: Identificador único (integer, chave primária)
- **nome**: Nome do template (string)
- **publico_alvo**: Público alvo do template (string)
- **semestre**: Semestre do template (string)
- **created_at**: Data de criação (datetime)
- **updated_at**: Data de atualização (datetime)

9. Turma

- **id**: Identificador único (integer, chave primária)
- **codigo**: Código da turma (string)
- **semestre**: Semestre da turma (string)
- **horario**: Horário da turma (string)
- **materium_id**: Chave estrangeira para Materium (integer)
- **created_at**: Data de criação (datetime)
- **updated_at**: Data de atualização (datetime)

10. User

- **id**: Identificador único (integer, chave primária)
- **matricula**: Matrícula do usuário (string)
- **email**: Email do usuário (string)
- **nome**: Nome do usuário (string)
- **role**: Papel do usuário (string)
- **password_digest**: Hash da senha (string)
- **created_at**: Data de criação (datetime)
- **updated_at**: Data de atualização (datetime)

Diagrama ER

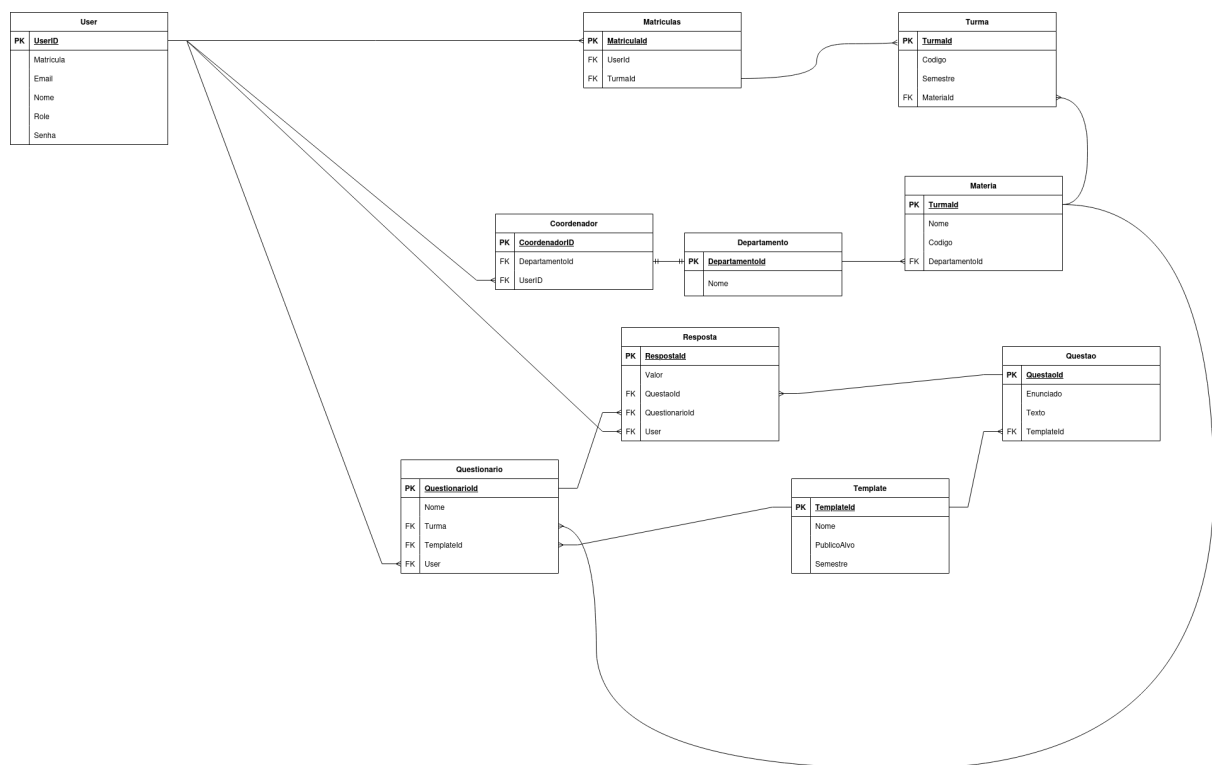


Figure 1: Diagrama Entidade-Relacionamento do Sistema CAMAAR

Modelo Relacional UML

Abaixo está o modelo relacional UML que descreve a estrutura de dados da aplicação **CAMAAR**, utilizamos praticamente o mesmo UML disponibilizado pelo monitor, com algumas modificações para se adequar ao nosso sistema.:

Welcome,

student

admin

professeur

Type de l'utilisateur

Signaler

OK, à gauche

Sign Up

Type des comptes :

☒ étudiant

☐ professeur

S'inscrire

Rester en

[Homepage](#)
[Perfil](#)
[Logout](#)



Profil
 Name: Alexy
 Registration: 2018/01
 Email: alexy@gmail.com
 Phone: 000000

Avaliação da disciplina	Avaliação da disciplina	Avaliação da disciplina
2018/2	2018/2	2018/2
8,000	8,000	8,000
ANACARDIO DE JARDIM	ANACARDIO DE JARDIM	ANACARDIO DE JARDIM

[Ver a resposta este questionário](#)

Homepage	Perfil	Logout
<h2>Avaliação da disciplina</h2> <p>Responda com cuidado, não é possível editar !</p> <p>Enunciado:</p> <p>Qual a sua opinião sobre a importância dos tópicos ministrados ?</p> <p> <input type="radio"/> Muito Importante <input type="radio"/> Importante <input type="radio"/> Médio <input type="radio"/> Pouco Importante <input type="radio"/> Irrelevante </p> <p>Resposta</p> <p><input type="text"/></p> <p>Enunciado:</p> <p>O que você achou do método utilizado pelo professor nas aulas ministradas ?</p> <p> <input type="radio"/> Muito bom <input type="radio"/> Bom <input type="radio"/> Médio <input type="radio"/> Ruim <input type="radio"/> Muito Ruim </p> <p>Resposta</p> <p><input type="text"/></p> <p>Ver resposta</p>		

[Homepage](#) [Profile](#) [Logout](#)

Editing user

Email

New Password (optional)

Confirm New Password (optional)

Available Pickups: [Previous](#) [Next](#) [Reset](#) [View all pickups on this order details](#)

[Update user](#)

[Show this user](#) / [Back](#)

Homepage	Perfil	Login
----------	--------	-------

Matriculas: 141128

Email: klay@gmail.com

Name: Klay

Beta index:

Password digest: E1aF12BCDXYt4S7bchwhbl qfJZGSLNvYjhdWkqBQl.akG7yb7pfTs

Matriculas

Colégio/Matricula	Materia
T.A. - 2012.1	BANCOS DE DADOS
T.A. - 2012.1	ENGENHARIA DE SOFTWARE
T.A. - 2012.1	PROGRAMAÇÃO CONCORRENTE

[Grá. Bk.user / Bkx](#)

[Homepage](#) | [Profile](#) | [Logout](#)

Editing user

Email

New Password (optional)

Confirm New Password (optional)

Profile Picture:

[Show this user](#) | [Back](#)

Importar datos	Editar templates	Ejecutar Formularios	Users	Resultados	Perfil	Logout
<div> <div>Profile</div> <div>  </div> <div> <p>Nome: doggy</p> <p>Registration: 10/10/2015</p> <p>Email: doggy@gmail.com</p> <p>Role: admin</p> <p>Coordenadas: 0,0</p> </div> </div>	<div> <div>Avaliacao da disciplina</div> <div> <p>000.0</p> <p>0000</p> <p>00000000000000000000</p> </div> </div>	<div> <div>Avaliacao da disciplina</div> <div> <p>000.0</p> <p>0000</p> <p>00000000000000000000</p> </div> </div>	<div> <div>Avaliacao da disciplina</div> <div> <p>000.0</p> <p>0000</p> <p>00000000000000000000</p> </div> </div>			



Figure 10: Importando Dados

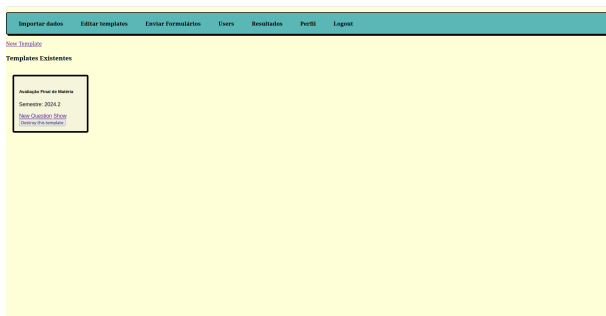


Figure 11: Gerenciamento de Templates



Figure 12: Criação de Template



Figure 13: Enviar formulários page

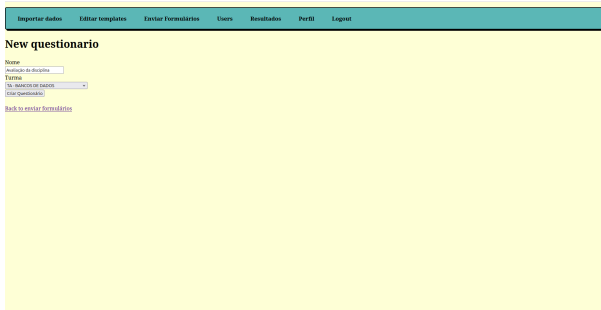


Figure 14: Criando questionario

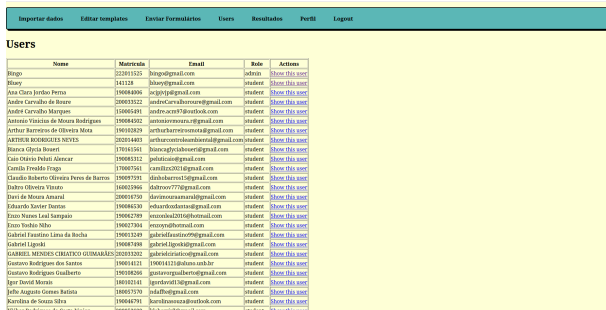


Figure 15: Tela de todos os usuários



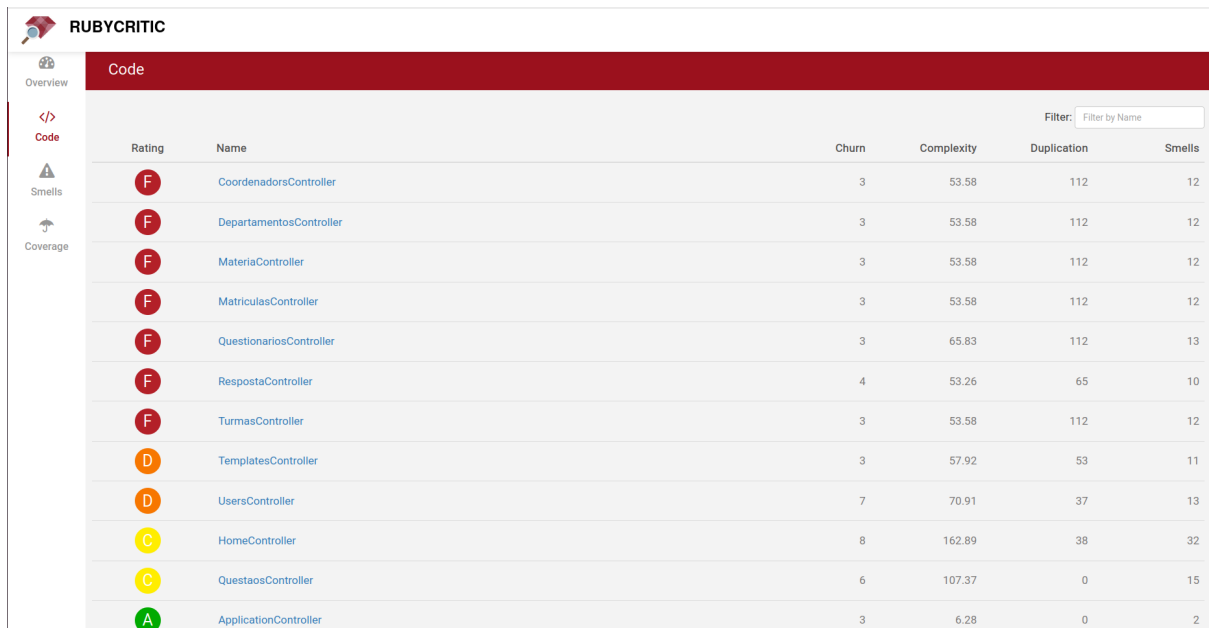
Figure 16: Editando um user sendo admin



Figure 17: Visualizando respostas

Refatoração

Para a refatoração utilizamos as ferramentas Rubycritic e Rubocop, que nos ajudaram a identificar e corrigir problemas de código e melhorar a qualidade do código.



The screenshot shows the Rubycritic web interface. On the left is a sidebar with navigation links: Overview, Code, Smells, and Coverage. The main area is titled 'Code' and contains a table of controllers. The table has columns for Rating, Name, Churn, Complexity, Duplication, and Smells. A search filter is available at the top right of the table.

Rating	Name	Churn	Complexity	Duplication	Smells
F	CoordenadoresController	3	53.58	112	12
F	DepartamentosController	3	53.58	112	12
F	MateriaController	3	53.58	112	12
F	MatriculasController	3	53.58	112	12
F	QuestionariosController	3	65.83	112	13
F	RespostaController	4	53.26	65	10
F	TurmasController	3	53.58	112	12
D	TemplatesController	3	57.92	53	11
D	UsersController	7	70.91	37	13
C	HomeController	8	162.89	38	32
C	QuestaosController	6	107.37	0	15
A	ApplicationController	3	6.28	0	2

Figure 18: Rubycritic antes das refatorações

Vários controllers possuíam métodos muito similares, principalmente os métodos create, update e destroy. Para resolver isso, criamos um módulo chamado CrudActions, que contém os métodos create, update e destroy, e incluímos esse módulo na maioria dos controllers, já que alguns possuíam métodos create/update/destroy um pouco mais complexos. Segue exemplo da controller MatriculasController a refatoração:

```

# app/controllers/matriculas_controller.rb
class MatriculasController < ApplicationController
  def index
    @matriculas = Matricula.all
  end

  def new
    @matricula = Matricula.new
  end

  def edit
  end

  def create
    @matricula = Matricula.new(matricula_params)
    respond_to do |format|
      if @matricula.save
        format.html { redirect_to @matricula }
        format.json { render :show, status: :created, location: @matricula }
      else
        format.html { render :new, status: :unprocessable_entity }
        format.json { render json: @matricula.errors, status: :unprocessable_entity }
      end
    end
  end

  def update
    respond_to do |format|
      if @matricula.update(matricula_params)
        format.html { redirect_to @matricula }
        format.json { render :show, status: :ok, location: @matricula }
      else
        format.html { render :edit, status: :unprocessable_entity }
        format.json { render json: @matricula.errors, status: :unprocessable_entity }
      end
    end
  end

  def destroy
    @matricula.destroy!
    respond_to do |format|
      format.html { redirect_to matriculas_path, status: :see_other }
      format.json { head :no_content }
    end
  end

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

```



```

def set_matricula
  @matricula = Matricula.find(params.expect(:id))
end

# Only allow a list of trusted parameters through.
def matricula_params
  params.expect(matricula: [ :user_id, :turma_id ])
end
end

```

Após a refatoração, o controller `MatriculasController` ficou assim:

```

# Controller responsável pelo gerenciamento de matrículas no sistema
class MatriculasController < ApplicationController
  include CrudActions

  before_action :set_matricula, only: %i[ show edit update destroy ]

  def index
    @matriculas = Matricula.all
  end

  def show
  end

  def new
    @matricula = Matricula.new
  end

  # Utiliza @matricula definido pelo before_action
  def edit
  end

  private

    def resource_class
      Matricula
    end

    def resource_params
      params.require(:matricula).permit(:user_id, :turma_id)
    end

    def set_matricula
      @matricula = Matricula.find(params[:id])
    end

    def after_destroy_path

```

```

    matriculas_path
  end

  def after_create_path
    matriculas_path
  end

  def after_update_path
    matriculas_path
  end
end

```

Idealmente todos os controllers deveriam estar verde. Porém, devido a complexidade de alguns controllers, não conseguimos refatorar todos.

RUBYCRITIC						
Code						
	Rating	Name	Churn	Complexity	Duplication	Smells
	C	CrudActions	2	68.8	41	9
	C	QuestaosController	8	108.87	32	16
	C	TemplatesController	5	36.49	41	5
	C	UsersController	10	53.16	16	7
	B	MemberImportService	2	59.84	0	9
	B	HomeController	10	89.83	0	12
	B	TurmasController	5	36.35	16	5
	A	ApplicationController	4	6.28	0	1
	A	CoordenadorsController	5	13.16	0	0
	A	DepartamentosController	5	13.16	0	0
	A	MateriaController	5	13.16	0	0
	A		5	13.16	0	0

Figure 19: Rubycritic após refatorações

Cobertura de Testes

Para a cobertura de testes, utilizamos a gem SimpleCov, que nos ajudou a identificar quais partes do código não estavam cobertas por testes.

All Files (54.44%)

Controllers (53.16%)

Channels (100.0%)

Models (97.14%)

Mailers (0.0%)

Helpers (0.0%)

Jobs (0.0%)

Libraries (100.0%)

Generated about 2 hours ago

Controllers (53.16% covered at 0.68 hits/line)

12 files in total.
427 relevant lines, 227 lines covered and 200 lines missed. (53.16%)

Search:

File	% covered [▲]	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/controllers/home_controller.rb	20.97 %	150	62	13	49	0.21
app/controllers/questaos_controller.rb	26.92 %	108	52	14	38	0.27
app/controllers/resposta_controller.rb	54.05 %	76	37	20	17	0.54
app/controllers/coordenadores_controller.rb	57.58 %	70	33	19	14	0.61
app/controllers/materia_controller.rb	57.58 %	70	33	19	14	0.61
app/controllers/matriculas_controller.rb	57.58 %	70	33	19	14	0.61
app/controllers/turmas_controller.rb	57.58 %	70	33	19	14	0.61
app/controllers/questionarios_controller.rb	61.11 %	73	36	22	14	0.64
app/controllers/departamentos_controller.rb	72.73 %	70	33	24	9	1.18
app/controllers/users_controller.rb	75.68 %	77	37	28	9	1.32
app/controllers/templates_controller.rb	78.79 %	70	33	26	7	1.42
app/controllers/application_controller.rb	80.00 %	9	5	4	1	0.80

Showing 1 to 12 of 12 entries

Figure 20: Cobertura de testes

All Files (54.44%) Controllers (53.16%) Channels (100.0%) Models (87.14%) Mailers (0.0%) Helpers (0.0%) Jobs (0.0%) Libraries (100.0%)							
Generated about 2 hours ago							
Models (87.14% covered at 0.97 hits/line)							
13 files in total. 70 relevant lines, 61 lines covered and 9 lines missed. (87.14%)							
Search: <input type="text"/>							
File	% covered ^A	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line	
app/models/materia.rb	0.00 %	4	4	0	4	0.00	
app/models/user.rb	81.48 %	43	27	22	5	1.07	
app/models/application_record.rb	100.00 %	3	2	2	0	1.00	
app/models/coordenador.rb	100.00 %	4	3	3	0	1.00	
app/models/departamento.rb	100.00 %	3	2	2	0	1.00	
app/models/materium.rb	100.00 %	4	3	3	0	1.00	
app/models/matricula.rb	100.00 %	4	3	3	0	1.00	
app/models/questao.rb	100.00 %	6	4	4	0	1.00	
app/models/questionario.rb	100.00 %	6	5	5	0	1.00	
app/models/resposta.rb	100.00 %	5	4	4	0	1.00	
app/models/respostum.rb	100.00 %	4	3	3	0	1.00	
app/models/template.rb	100.00 %	8	6	6	0	1.00	
app/models/turma.rb	100.00 %	5	4	4	0	1.00	

Figure 21: Cobertura de testes

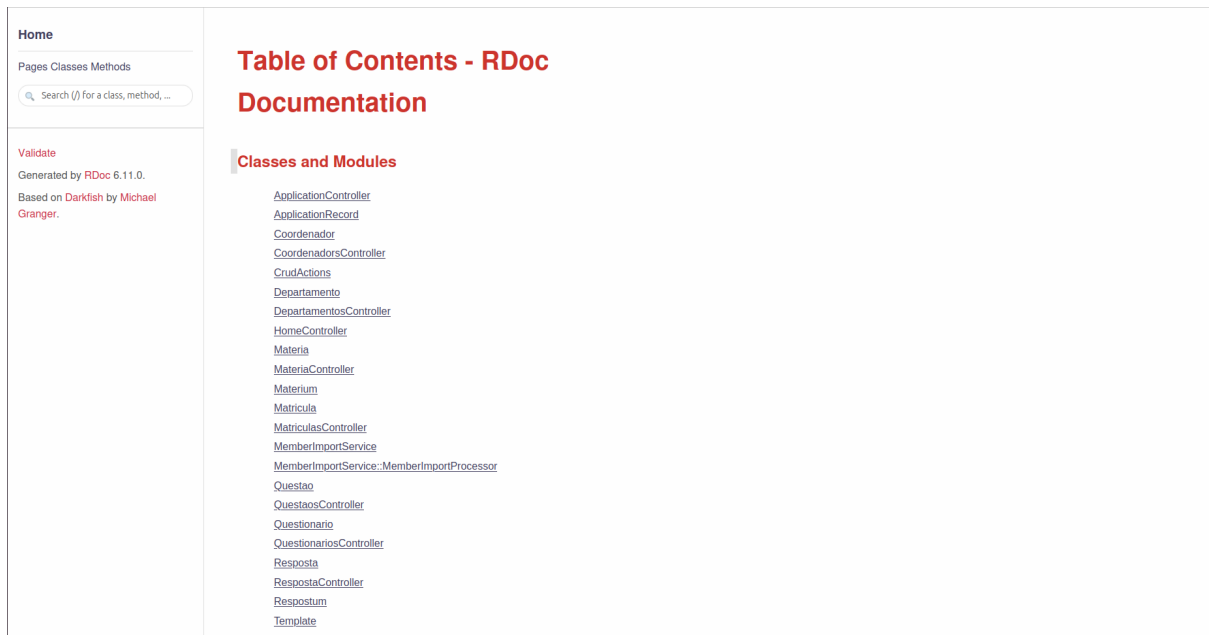


Figure 22: Documentação automática

Documentação automática

Para a documentação automática, utilizamos a gem `rdoc`, que nos ajudou a gerar a documentação automática do código.

Conclusão

Durante o desenvolvimento do projeto, nos deparamos com diversos impeditivos que atrasaram o desenvolvimento do projeto. Vários membros do grupo estavam com muitas atividades da faculdade e do trabalho, o que dificultou a comunicação e a divisão de tarefas. Além disso, consideramos que a parte mais complexa do projeto foi a criação de testes tanto no Rspec quanto no Cucumber/Capybara, pois a maioria dos membros do grupo não tinham experiência com tais ferramentas. Por fim, acreditamos que o projeto foi um sucesso, parcial, pois conseguimos implementar todas as funcionalidades propostas. Porém, acreditamos que pecamos na parte da criação de testes e na refatoração do código, que poderiam ter sido feitas de forma mais eficiente.