

**DELPo**

**André Luiz  
Abdalla  
Silveira**

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

# **MAC 0499 – Dicionário Etimológico da Língua Portuguesa**

**André Luiz Abdalla Silveira**

Instituto de Matemática e Estatística  
Universidade de São Paulo

3 de dezembro de 2019



IME-USP

DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

## 1 Introdução

## 2 Propostas

- Interface de usuário
- Testes
- Refatoração

## 3 Resultados

- API
- Interface
- Balanco Geral

## 4 Aprendizados

## 5 Bibliografia



IME-USP

DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

## 1 Introdução

## 2 Propostas

- Interface de usuário
- Testes
- Refatoração

## 3 Resultados

- API
- Interface
- Balanco Geral

## 4 Aprendizados

## 5 Bibliografia

# O que é o DELPo



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- material de estudo de um TCC no ano passado

# O que é o DELPo



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- material de estudo de um TCC no ano passado
- acrônimo de Dicionário Etimológico da Língua Portuguesa

# O que é o DELPo



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- material de estudo de um TCC no ano passado
- acrônimo de Dicionário Etimológico da Língua Portuguesa
- projeto criado em 2012 pelo Prof. Dr. Mário Eduardo Viaro (DLCV – FFLCH-USP)

# O que é o DELPo



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- material de estudo de um TCC no ano passado
- acrônimo de Dicionário Etimológico da Língua Portuguesa
- projeto criado em 2012 pelo Prof. Dr. Mário Eduardo Viaro (DLCV – FFLCH-USP)
- destinado a facilitar o estudo de etimologia

# O que é o DELPo



## DELPo

André Luiz  
Abdalla  
Silveira

### Introdução

### Propostas

Interface de usuário

Testes

Refatoração

### Resultados

API

Interface

Balanco Geral

### Aprendizados

### Bibliografia

- material de estudo de um TCC no ano passado
- acrônimo de Dicionário Etimológico da Língua Portuguesa
- projeto criado em 2012 pelo Prof. Dr. Mário Eduardo Viaro (DLCV – FFLCH-USP)
- destinado a facilitar o estudo de etimologia
- plataforma processa textos recebidos e alimenta a base de dados



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- Moedor – principal e foco da refatoração

## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- Moedor – principal e foco da refatoração
- Metapasmador

## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- Moedor – principal e foco da refatoração
- Metapasmador
- Analisador de frequência ao longo dos anos

## DELPo

André Luiz  
Abdalla  
Silveira

### Introdução

### Propostas

Interface de usuário

Testes

Refatoração

### Resultados

API

Interface

Balanco Geral

### Aprendizados

### Bibliografia

## 1 Introdução

## 2 Propostas

- Interface de usuário
- Testes
- Refatoração

## 3 Resultados

- API
- Interface
- Balanço Geral

## 4 Aprendizados

## 5 Bibliografia

# A camada de apresentação



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

- A aplicação funcionava, mas a interface feita estava sem estilização alguma

# A camada de apresentação



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

- A aplicação funcionava, mas a interface feita estava sem estilização alguma
- ActiveViewer (Rails) → Nuxt.js
  - facilidade de uso

# A camada de apresentação



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

- A aplicação funcionava, mas a interface feita estava sem estilização alguma
- ActiveViewer (Rails) → Nuxt.js
  - facilidade de uso
  - flexibilidade

# A camada de apresentação



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

- A aplicação funcionava, mas a interface feita estava sem estilização alguma
- ActiveViewer (Rails) → Nuxt.js
  - facilidade de uso
  - flexibilidade
  - menos “poluição” no controlador



# A camada de apresentação



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

- A aplicação funcionava, mas a interface feita estava sem estilização alguma
- ActiveViewer (Rails) → Nuxt.js
  - facilidade de uso
  - flexibilidade
  - menos “poluição” no controlador
- Aplicação Rails no modo API

# Testando as aplicações



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

### Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- permitem verificar o funcionamento (ou não) do código

# Testando as aplicações



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- permitem verificar o funcionamento (ou não) do código
- pressupõe que o testador conheça o código

# Testando as aplicações



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

- permitem verificar o funcionamento (ou não) do código
- pressupõe que o testador conheça o código
- O que se pode verificar?
  - Comportamento da aplicação — Rails (testes em RSpec)

# Testando as aplicações



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

- permitem verificar o funcionamento (ou não) do código
- pressupõe que o testador conheça o código
- O que se pode verificar?
  - Comportamento da aplicação — Rails (testes em RSpec)
  - Integração de componentes — Nuxt (testes em Jest)

# Testando as aplicações



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

- permitem verificar o funcionamento (ou não) do código
- pressupõe que o testador conheça o código
- O que se pode verificar?
  - Comportamento da aplicação — Rails (testes em RSpec)
  - Integração de componentes — Nuxt (testes em Jest)
- Cobertura de testes na aplicação Rails: 72%

# Testando as aplicações



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

- permitem verificar o funcionamento (ou não) do código
- pressupõe que o testador conheça o código
- O que se pode verificar?
  - Comportamento da aplicação — Rails (testes em RSpec)
  - Integração de componentes — Nuxt (testes em Jest)
- Cobertura de testes na aplicação Rails: 72%
- Cobertura de testes na aplicação Nuxt: Não haviam testes

# Melhorando o código



## DELPo

André Luiz  
Abdalla  
Silveira

- única das propostas a não estar na apresentação inicial

## Introdução

## Propostas

Interface de usuário

Testes

**Refatoração**

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia



# Melhorando o código



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

## Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- única das propostas a não estar na apresentação inicial
- “Cheiros” do código nunca foram tão incômodos

# Melhorando o código



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

## Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- única das propostas a não estar na apresentação inicial
- “Cheiros” do código nunca foram tão incômodos
- Motivos prováveis

# Melhorando o código



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- única das propostas a não estar na apresentação inicial
- “Cheiros” do código nunca foram tão incômodos
- Motivos prováveis
  - Pressa para cumprir prazos

# Melhorando o código



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- única das propostas a não estar na apresentação inicial
- “Cheiros” do código nunca foram tão incômodos
- Motivos prováveis
  - Pressa para cumprir prazos
  - Falta de experiência de mercado com Rails

# Melhorando o código



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- única das propostas a não estar na apresentação inicial
- “Cheiros” do código nunca foram tão incômodos
- Motivos prováveis
  - Pressa para cumprir prazos
  - Falta de experiência de mercado com Rails
- refatorar o código não quer dizer que o código é ruim

# Melhorando o código



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- única das propostas a não estar na apresentação inicial
- “Cheiros” do código nunca foram tão incômodos
- Motivos prováveis
  - Pressa para cumprir prazos
  - Falta de experiência de mercado com Rails
- refatorar o código não quer dizer que o código é ruim
- é benéfico à aplicação

# Melhorando o código



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- única das propostas a não estar na apresentação inicial
- “Cheiros” do código nunca foram tão incômodos
- Motivos prováveis
  - Pressa para cumprir prazos
  - Falta de experiência de mercado com Rails
- refatorar o código não quer dizer que o código é ruim
- é benéfico à aplicação
  - tornou-a mais organizada, mais inteligível, e por isso, mais escalável

# Melhorando o código



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

- única das propostas a não estar na apresentação inicial
- “Cheiros” do código nunca foram tão incômodos
- Motivos prováveis
  - Pressa para cumprir prazos
  - Falta de experiência de mercado com Rails
- refatorar o código não quer dizer que o código é ruim
- é benéfico à aplicação
  - tornou-a mais organizada, mais inteligível, e por isso, mais escalável
  - exemplos: criação de *helpers* e uso de serializadores (transformam conjuntos de dados em um JSON)



# Exemplo de refatoração



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

```
# recebe uma string de data e retorna para data no tipo aaaa-mm-dd
def trataData (dat)
  # remove o c da incerteza
  dat.gsub!(/c/, '')
  #data completa
  if match = dat.match(/(\d+)\D(\d+)\D(\d+)/)
    s1, s2, s3 = match.captures
    return s3 + "." + s2 + "." + s1
  elsif match = dat.match(/^(?!\d{1,2})\D(\d+)/)
    # sem dia, 05/1811, por exemplo
    s1, s2 = match.captures
    return ultimoDia(s1.to_i, s2.to_i)
  elsif match = dat.match(/^(?!\d+)\D(\d+)/)
    #2000-2001, por exemplo
    s1 = match.captures
    return s1 + "-12-31"
  end
  #XV-XVII por exemplo
  elsif match = dat.match(/(X{0,3}(IV|IX|IV|V|I{0,3}))- (X{0,3}(IV|IX|IV|V|I{0,3}))/)
    s1, s2, s3 = match.captures
    ans = trataNumeral(s1) > trataNumeral(s3) ? trataNumeral(s1) : trataNumeral(s3)
    return ans + "-01-01"
  end
  #2000 por exemplo
  elsif match = dat.match(/(?[0-9]{1,4})/)
    return dat + "-12-31"
  end
  #XVII, por exemplo
  else
    return trataNumeral(dat).to_s + "-01-01" #1200-01-01 é o último dia do século XII
  end
end
```

```
def ultimo_dia_do_seculo (n)
  Date.new(Roman::Numerals.to_decimal(n) * 100, -1, -1)
end

def eh_numerico? (ano)
  Integer(ano) != nil rescue false
end

def ajusta_data_com_dois_elementos (elemento_1, elemento_2)
  return ultimo_dia_do_seculo(elemento_2) unless eh_numerico?(elemento_2)
  begin
    data = Date.new(elemento_1.to_i, elemento_2.to_i, -1)
  rescue ArgumentError
    data = Date.new(elemento_2.to_i, -1, -1)
  ensure
    return data
  end
end

def ajusta_data_com_um_elemento(elemento_1)
  return ultimo_dia_do_seculo(elemento_1) unless eh_numerico?(elemento_1)
  Date.new(elemento_1.to_i, -1, -1)
end

def remove_incerteza_data
  data.gsub!(/c/, '')
end

def ajusta_data
  remove_incerteza(data)
  data_dividida = data.split('.')
  if data_dividida.length == 3
    Date.new(data_dividida[0].to_i, data_dividida[1].to_i, data_dividida[2].to_i)
  elsif data_dividida.length == 2
    ajusta_data_com_dois_elementos[data_dividida[0], data_dividida[1]]
  else
    ajusta_data_com_um_elemento(data_dividida[0])
  end
end
```

(a) Antes

(b) Depois

Figura: Exemplo tirado do DataHelper

# O que foi feito?



## DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

## 1 Introdução

## 2 Propostas

- Interface de usuário
- Testes
- Refatoração

## 3 Resultados

- API
- Interface
- Balanco Geral

## 4 Aprendizados

## 5 Bibliografia

## DELPo

André Luiz  
Abdalla  
Silveira

### Introdução

### Propostas

Interface de usuário

Testes

Refatoração

### Resultados

**API**

Interface

Balanco Geral

### Aprendizados

### Bibliografia

- Foco em “despoluir” os controladores

## DELPo

André Luiz  
Abdalla  
Silveira

- Foco em “despoluir” os controladores
- Uso da gema `active_model_serializers`

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

**API**

Interface

Balanco Geral

## Aprendizados

## Bibliografia

## DELPo

André Luiz  
Abdalla  
Silveira

- Foco em “despoluir” os controladores
- Uso da gema `active_model_serializers`

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanço Geral

## Aprendizados

## Bibliografia

```
class ObrasController < ApplicationController
  before_action :set_obra, only: [:show, :edit, :update, :destroy]
  load_and_authorize_resource
  # GET /obras
  # GET /obras.json
  def index
    @obras = Obra.all
    autores = Array.new(@obras.length)
    editoras = Array.new(@obras.length)
    i = 0
    @obras.each do |obra|
      aut = Autor.find_by id: obra["autorint_id"]
      edi = Editora.find_by id: obra["editora_id"]
      if aut == nil
        autores[i] = ""
      else
        autores[i] = aut["nome"]
      end
      if edi == nil
        editoras[i] = ""
      else
        editoras[i] = edi["nome"]
      end
      i += 1
    end
    @autores = autores
    @editoras = editoras
  end
end
```

(a) Antes

```
class ObrasController < ApplicationController
  before_action :set_obra, only: [:show, :update, :destroy]
  before_action :authenticate_usuario!, only: [:create, :update, :destroy]

  # GET /obras
  # GET /obras.json
  def index
    render json: Obra.all
  end
end
```

(b) Depois

**Figura:** Exemplo tirado do ObrasController

# Serializer



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

```
class ObrasSerializer < ActiveModel::Serializer
  include ObrasHelper
  attributes *attrs :id, :texto_dia, :public_data, :texto_titulo, :public_titulo,
    :texto_local, :localiza, :edicao_numero, :edicao_volume, :edicao_tipo,
    :cota, :univdisc, :genero, :suporte, :concedente, :comentarios, :organizador

  # has_many :autorias
  has_many name :autores, through: :autorias

  def texto_dia
    format_data object.texto_data
  end
end
```

Figura: ObrasSerializer

# Requisição



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

```
GET localhost:4000/obras Send 200 OK TIME 297 ms SIZE 330 B

JSON Auth Query Header Docs Preview Header Cookie Timeline

1 {}
2

1 [
2   {
3     "id": 1,
4     "texto_dia": "20/ 2/2020",
5     "public_data": "2019",
6     "texto_titulo": "Uma super Obra",
7     "public_titulo": null,
8     "texto_local": null,
9     "localiza": null,
10    "edicao_numero": null,
11    "edicao_volume": null,
12    "edicao_tipo": null,
13    "cota": null,
14    "univdisc": null,
15    "genero": null,
16    "suporte": null,
17    "concedente": null,
18    "comentarios": null,
19    "organizador": null,
20    "autores": []
21  }
22 ]
```

**Figura:** Resposta à requisição GET localhost:4000/obras

# Camada de Apresentação



**DELPo**

**André Luiz  
Abdalla  
Silveira**

- Conteúdos estáticos já estão prontos

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

**Interface**

Balanco Geral

Aprendizados

Bibliografia



# Camada de Apresentação



DELPo

André Luiz  
Abdalla  
Silveira

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

Balanco Geral

Aprendizados

Bibliografia

- Conteúdos estáticos já estão prontos
- Uso da ferramenta Vuetify. Um plug-in feito para Vue.js e seus derivados e que usa Material Design.
  - Segundo o site <https://material.io/design/introduction/>, é um paradigma que une princípios clássicos do bom design com a inovação da tecnologia e ciência.

## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- Conteúdos estáticos já estão prontos
- Uso da ferramenta Vuetify. Um plug-in feito para Vue.js e seus derivados e que usa Material Design.
  - Segundo o site <https://material.io/design/introduction/>, é um paradigma que une princípios clássicos do bom design com a inovação da tecnologia e ciência.
- Foco: interligar com a API além do sistema de login

## Camada de Apresentação



DELPo

**André Luiz  
Abdalla  
Silveira**

## Introdução

## Propostas

## Interface de usuário

## Testes

## Refatoração

## Resultados

API

## Interface

## Balanço Geral

## Aprendizados

## Bibliografia

- Conteúdos estáticos já estão prontos
- Uso da ferramenta Vuetify. Um plug-in feito para Vue.js e seus derivados e que usa Material Design.
  - Segundo o site <https://material.io/design/introduction/>, é um paradigma que une princípios clássicos do bom design com a inovação da tecnologia e ciência.
- Foco: interligar com a API além do sistema de login



(a) /home

[illegible]

(b) /equipe

**Figura:** Captura de telas feitas a partir da interface



## DELPo

André Luiz  
Abdalla  
Silveira

- Concluídas
  - Refatorar o banco de dados

Introdução

Propostas

Interface de usuário

Testes

Refatoração

Resultados

API

Interface

**Balanco Geral**

Aprendizados

Bibliografia

## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

**Balanco Geral**

## Aprendizados

## Bibliografia

- Concluídas
  - Refatorar o banco de dados
  - Refatoração dos modelos e controladores

## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

**Balanco Geral**

## Aprendizados

## Bibliografia

- Concluídas
  - Refatorar o banco de dados
  - Refatoração dos modelos e controladores
  - Modularização das classes e interfaces

## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

**Balanco Geral**

## Aprendizados

## Bibliografia

- Concluídas
  - Refatorar o banco de dados
  - Refatoração dos modelos e controladores
  - Modularização das classes e interfaces
- Em Andamento
  - Fazer uma interface de usuário agradável (60%)

## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

**Balanco Geral**

## Aprendizados

## Bibliografia

- Concluídas
  - Refatorar o banco de dados
  - Refatoração dos modelos e controladores
  - Modularização das classes e interfaces
- Em Andamento
  - Fazer uma interface de usuário agradável (60%)
  - Fazer os testes no back (73%)



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- Concluídas
  - Refatorar o banco de dados
  - Refatoração dos modelos e controladores
  - Modularização das classes e interfaces
- Em Andamento
  - Fazer uma interface de usuário agradável (60%)
  - Fazer os testes no back (73%)
  - Refatoração do moedor (70%)

## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- Concluídas
  - Refatorar o banco de dados
  - Refatoração dos modelos e controladores
  - Modularização das classes e interfaces
- Em Andamento
  - Fazer uma interface de usuário agradável (60%)
  - Fazer os testes no back (73%)
  - Refatoração do moedor (70%)
- Não Iniciadas
  - Adaptação dos dados do banco de dados antigo

## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- Concluídas
  - Refatorar o banco de dados
  - Refatoração dos modelos e controladores
  - Modularização das classes e interfaces
- Em Andamento
  - Fazer uma interface de usuário agradável (60%)
  - Fazer os testes no back (73%)
  - Refatoração do moedor (70%)
- Não Iniciadas
  - Adaptação dos dados do banco de dados antigo
  - Fazer os testes do front

## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- Concluídas
  - Refatorar o banco de dados
  - Refatoração dos modelos e controladores
  - Modularização das classes e interfaces
- Em Andamento
  - Fazer uma interface de usuário agradável (60%)
  - Fazer os testes no back (73%)
  - Refatoração do moedor (70%)
- Não Iniciadas
  - Adaptação dos dados do banco de dados antigo
  - Fazer os testes do front
- Observação: As porcentagens na coluna do meio são uma estimativa do grau de conclusão da tarefa

# O que aprendi?



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- A importância do bom código

# O que aprendi?



## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- A importância do bom código
- Equilíbrio entre a vontade de fazer o melhor e terminar determinadas tarefas

## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanco Geral

## Aprendizados

## Bibliografia

- O dicionário etimológico da língua portuguesa (delpo): conceitos de metalema, hemilema, hiperlema e ultralema — Mário Eduardo Viaro — 2017
- Boas práticas — Daniel Schmitz — 2019
- Refatoração do Projeto Delpo — Adriano Tetsuaki Ogawa Santin, Luiz Fernando Antonelli Galati e Mauricio Luiz Abreu Cardoso — 2018
- O uso do dicionário de língua como instrumento didático no ensino de língua portuguesa para alunos surdos: em busca de um bilinguismo funcional — Barbara Neves Salviano — 2014
- Clean code — Wojtek Lukaszuk — 2018
- Ruby Cookbook — Lucas Carlson e Leonard Richardson — 2009
- Manual do NEHiLP

## DELPo

André Luiz  
Abdalla  
Silveira

## Introdução

## Propostas

Interface de usuário

Testes

Refatoração

## Resultados

API

Interface

Balanço Geral

## Aprendizados

## Bibliografia

- [Ruby on Rails Guides](#)
- [Vue.js Docs](#)
- [Nuxt.JS Guide](#)
- [Jest Homepage](#)