



# **Ampliando o sistema: Integração de Atividades de Extensão no Sistema Acadêmico da UENF**

*Gabriel Almeida, Nathan de Oliveira, Mariana Cossetti, Enzo Marques,  
Enzo Alberoni - Annabell Tamariz*

**UENF - CCT - LCMAT - CC**

**30 de novembro de 2023**

Copyright © 2017-2023 Ausberto S. Castro Vera e Enzo Marques Souza

UENF - UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO

CCT - CENTRO DE CIÊNCIA E TECNOLOGIA

LCMAT - LABORATÓRIO DE MATEMÁTICAS

CC - CURSO DE CIÊNCIA DA COMPUTAÇÃO



## Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivo	1
1.2	Modelagem	2
<b>2</b>	<b>Progressão do Projeto</b>	<b>5</b>
2.0.1	Dia 16/11 UML e IDE	5
2.0.2	Dia 17/11 - Criação Login	5
2.0.3	Dia 18/11 Controllers de Atividades	6
2.0.4	Dia 19/11 Resolução de Bugs	6
2.0.5	Dia 20/11 Criação das páginas Aluno	7
2.0.6	Dia 21/11 ao 26/11 Validar as atividades	7
2.0.7	Dia 28/11 Histórico e finalização	8
<b>3</b>	<b>Conceitos de POO aplicados no código</b>	<b>9</b>
3.0.1	Classe	9
3.0.2	Herança	9
3.0.3	Métodos	9
3.0.4	Atributos	10
3.0.5	Encapsulamento	10
3.0.6	Variável de instância	10
<b>3.1</b>	<b>Atribuição de tarefas</b>	<b>11</b>
<b>4</b>	<b>Resultados</b>	<b>13</b>

---

<b>5</b>	<b>Conclusão e recomendação .....</b>	<b>19</b>
<b>5.1</b>	<b>Recomendações</b>	<b>19</b>



Atividades Curriculares de Extensão (ACE)	
Modalidade da Atividade	CH Máxima
ACE I – Projetos e Programas	200h
ACE II – Cursos e Eventos	200h
ACE III – Prestação de Serviços	200h
ACE IV – Disciplinas Extensionista	200h

Figura 1.1: Atividades de Extensão

No sistema, cada aluno tem seu perfil e possui suas atividades cadastradas em uma tabela exibindo todos os dados da atividade e seus status atual, é permitido cadastrar as atividades que foram feitas afim do coordenador valida-las. Caso a atividade seja validada, as horas vão ser contabilizadas.

## 1.2 Modelagem

Foi feita a primeira modelagem logo no início de outubro.

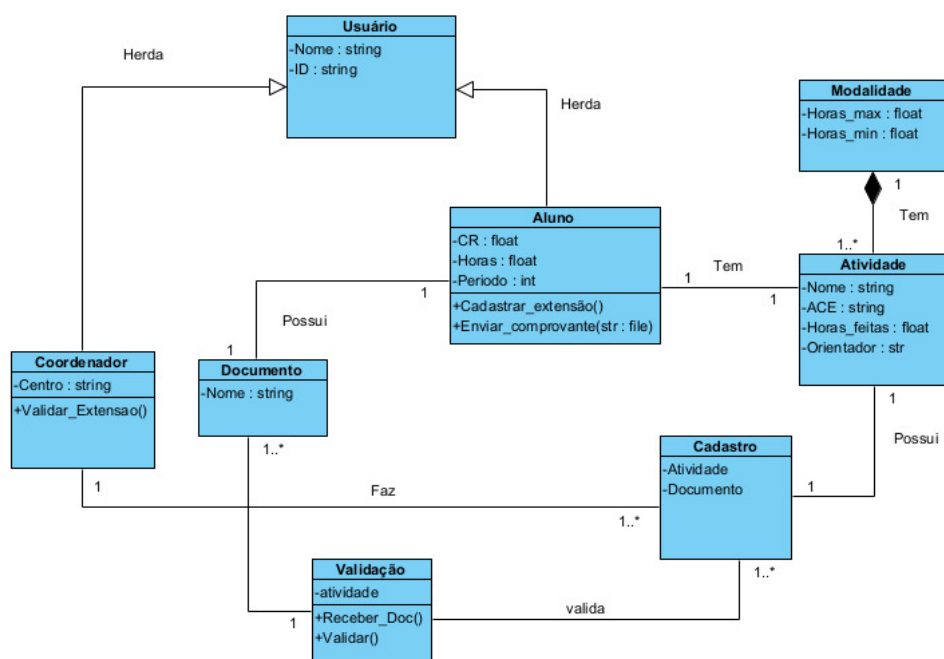


Figura 1.2: Primeira Modelagem



Haveria um usuário como classe pai de **Aluno** e **Coordenador**. Coordenador cadastraria as atividades, e o aluno escolheria a atividade dentre uma listas de possíveis. O aluno, no final, enviaria um documento comprovando que a atividade foi feita, e caberia ao coordenador validar. Decidimos abandonar essa ideia por estar mais complexo ao que deveria ser.

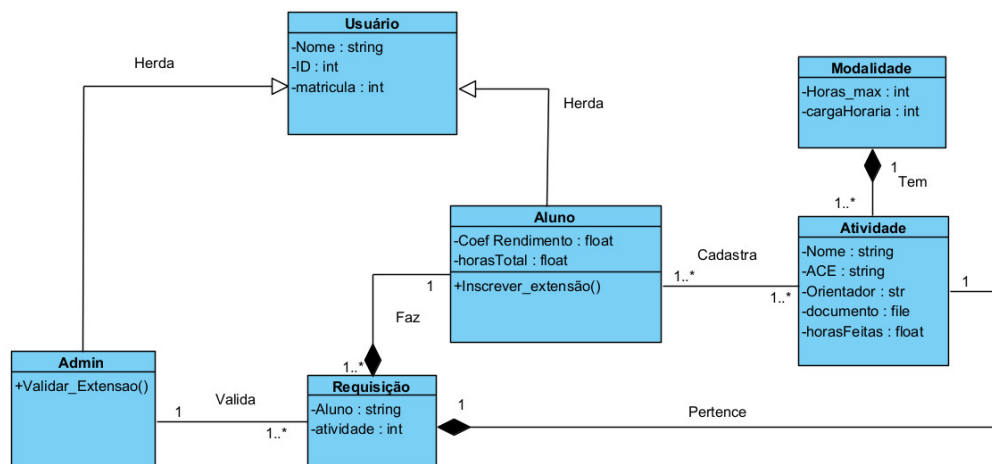


Figura 1.3: Segunda Modelagem

A segunda modelagem foi feita juntamente com os professores João e Rafaela. Decidimos reformular todo o projeto. Agora, a ideia é que o aluno cadastre a atividade após realizá-la, enviando um documento comprovando-a. O coordenador, por sua vez, iria validar se a atividade é válida ou não. Optamos por essa abordagem por ser mais simples de implementar.

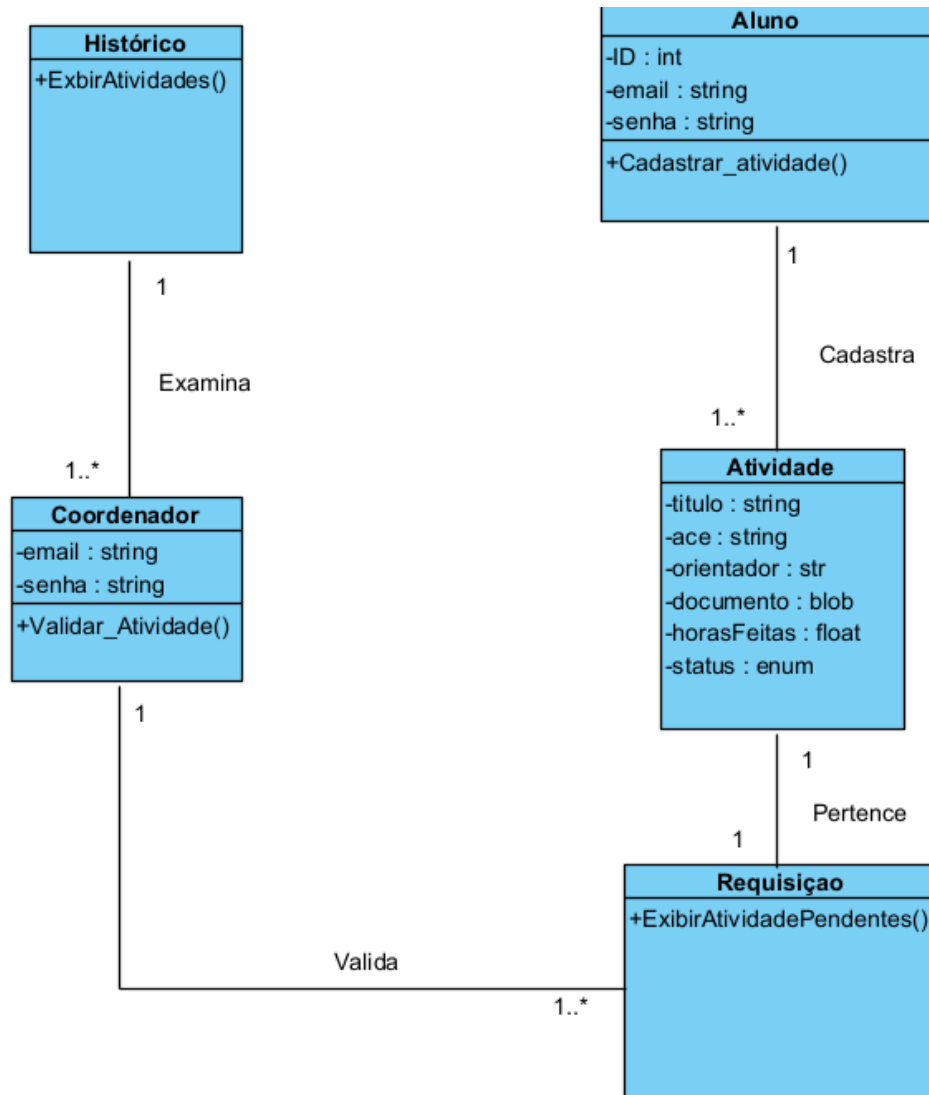


Figura 1.4: Enter Caption

A modelagem final foi ajustada de acordo com o que conseguimos implementar no código. Mantivemos o princípio de que o aluno cadastra as atividades após concluí-las. O coordenador valida essas atividades por meio de uma requisição que contém todas as atividades pendentes. Além disso, o coordenador pode visualizar o histórico de todas as atividades já solicitadas pelos alunos.





**Polimorfismo**, seria melhor. Por conta disso não foi possível criar a classe pai **Usuários**, o qual quebra outro princípio da orientação a objetos de **Herança**, pois **aluno** e **coordenador** iriam herdar atributos de **Usuários**.

Para melhorar a experiência do usuário, implementamos formulários de cadastro e login. Esses **forms** foram projetados para permitir que os usuários utilizem seus e-mails para logar no sistema. Cada usuário tem suas próprias atividades sendo diferenciados pelo **aluno ID**. Essa integração facilita o processo de autenticação, mantendo uma abordagem familiar aos usuários acadêmicos.

### 2.0.3 Dia 18/11 Controllers de Atividades

Neste dia, concentramos nossos esforços no desenvolvimento crucial dos **controllers** e **rotas** necessários para a gestão efetiva das atividades no sistema. Os **controllers** foram projetados para manipular as interações do usuário relacionadas às atividades.

Além disso, estabelecemos e configuramos a estrutura do banco de dados dedicada às atividades. A modelagem do banco de dados foi pensada de forma a permitir a representação das informações relacionadas às atividades, facilitando.

```
1      create_table "atividades", force: :cascade do |t|
2          t.string "titulo"
3          t.string "ace"
4          t.string "orientador"
5          t.binary "documento"
6          t.float "horasFeitas"
7          t.datetime "created_at", precision: 6, null: false
8          t.datetime "updated_at", precision: 6, null: false
9          t.integer "aluno_id"
10         t.string "status", default: "pending"
11         t.integer "coordenador_id"
12         t.index ["coordenador_id"], name:
13             "index_atividades_on_coordenador_id"
14     end
```

[

Acima, um trecho do banco de dados retirado do **schema.rb**. Essas foram as tabelas criadas no banco de dados de atividades.

No geral, o dia foi marcado por avanços significativos na implementação do sistema, com um foco especial na integração perfeita com o ambiente acadêmico existente e na resolução ativa de desafios técnicos.

### 2.0.4 Dia 19/11 Resolução de Bugs

No decorrer deste dia, concentramos nossos esforços na resolução de bugs identificados, continuando a trabalhar na estabilidade e confiabilidade do sistema. Além disso, enfrentamos desafios adicionais relacionados à utilização da IDE, buscando soluções para garantir um ambiente de desenvolvimento eficiente. O principal problema era que não estava sendo feito o login de alunos, mas descobrimos posteriormente que era um erro do Replit.

### 2.0.5 Dia 20/11 Criação das páginas Aluno

Implementamos caminhos específicos para os alunos dentro do sistema, onde eles têm a opção de cadastrar e editar as atividades realizadas. Após o login, o aluno é direcionado para uma **Página**, onde seleciona a aba **atividades**, apresentando uma lista das atividades realizadas com seus **status** e as opções de cadastrar ou editar.

Dentro da aba de cadastro, foi criado um formulário informando o título, modalidade, ACE, orientador e as horas feitas. Ao cadastrar, essa atividade era inserida no banco de dados do aluno com o status 'pending', aguardando a validação do coordenador.

Além do início do design do sistema, foram criadas inicialmente as páginas de **Login**, **Forgot your password?** para o **aluno** e **coordenador**, e a página de **Signup** apenas para o **aluno** com o auxílio do framework Bootstrap.

Essa etapa representa um avanço significativo, consolidando a contribuição do nosso sistema ao ambiente acadêmico da UENF e expandindo suas capacidades para melhor atender às necessidades dos alunos e coordenadores.

### 2.0.6 Dia 21/11 ao 26/11 Validar as atividades

Durante o período de 21 a 26 de novembro, nossa equipe dedicou esforços consideráveis à implementação da validação das atividades pelos administradores, com o objetivo de aprimorar a integridade e eficiência do sistema. Foi uma das partes mais complicadas no projeto; passamos dias tentando implementar o update, mas por alguns motivos, não atualizava no banco de dados. A ideia principal era criar um novo controller **requisição**; ele seria responsável por validar as atividades. Entretanto, não conseguimos aplicar na prática como queríamos. Foi criada a **requisicao**, mas toda a parte do update foi feita diretamente no **controller de atividade**. Após analisarmos, percebemos que nosso problema foi possivelmente a relação de associar requisição com atividades, e tivemos pouco contato com Rails.

```
1      class AtividadesController < ApplicationController
2
3          def edit
4              @atividade = Atividade.find(params[:id])
5          end
6
7          def update
8              @atividade = Atividade.find(params[:id])
9              if @atividade.update(atividade_params) &&
10                  current_aluno
11                  redirect_to atividades_path
12
13              elsif
14                  redirect_to requisicao_index_path
15              else
16                  render :edit, status: :unprocessable_entity
17              end
18          end
19      end
```

Aqui está como foi feito o update dentro do controller de atividades. Nossa ideia era apenas criar um botão com **Validar** ou **Rejeitar**, mas por algum motivo o botão não funcionava. Então, tivemos que partir para um formulário, onde são coletados todos os dados da atividade e permitir ao coordenador validar ou não. Ao entrar na página, o coordenador se depara com uma lista de todas as atividades com status pendentes.

O objetivo principal neste período foi possibilitar que o coordenador do sistema validasse as atividades submetidas pelos alunos. Essa funcionalidade é crucial para garantir a precisão e confiabilidade das informações registradas no sistema acadêmico.

### 2.0.7 Dia 28/11 Histórico e finalização

Decidimos criar um novo controller e banco de dados para os históricos, que estão vinculados às atividades. O banco de dados possui uma referência às atividades na tabela, proporcionando uma organização eficiente das informações. Esse histórico é acessado pelo coordenador e abrange todas as atividades já realizadas por todos os alunos.

Por fim, a parte do design do sistema das páginas que ainda não tinham nenhum estilo, estando apenas em HTML, foi finalizada. E implementamos a funcionalidade de soma das horas totais de todas as atividades validadas. Isso proporciona uma visão detalhada das horas acumuladas pelos alunos em suas atividades.





### 3.0.4 Atributos

Os atributos são características de um objeto, geralmente na programação definimos dentro da classe, mas no rails fazemos essa definição pelo banco de dados. Por exemplo, os atributos de atividades.

```
1 create_table "atividades", force: :cascade do |t|
2   t.string "titulo"
3   t.string "ace"
4   t.string "orientador"
5   t.binary "documento"
6   t.float "horasFeitas"
7   t.datetime "created_at", precision: 6, null: false
8   t.datetime "updated_at", precision: 6, null: false
9   t.integer "aluno_id"
10  t.string "status", default: "pending"
11  t.integer "coordenador_id"
12  t.index ["coordenador_id"], name: "
    index_atividades_on_coordenador_id"
13 end
```

### 3.0.5 Encapsulamento

O conceito de encapsulamento é permitir se um atributo ele pode ser público, privado ou protegido. O público permite acesso de todas as classes. Já o privado, ocorre limitação onde apenas a classe tem acesso a seus atributos, por fim, o protegido permite que na herança a classe filho consiga acessar os atributos do pai. Um exemplo de encapsulamento no código

```
1 private
2 def atividade_params
3   params.require(:atividade).permit(:titulo, :
      orientador, :ace, :horasFeitas, :documento, :
      status)
4 end
```

### 3.0.6 Variável de instância

As variáveis de instâncias são variáveis onde seu valor é específico ao objeto e não à classe. Em nosso código, utilizamos a variável **@atividade** para pegar todas atividades de cada aluno.

```
1 def index
2   @atividade = current_aluno.atividades.all
3 end
```

### 3.1 Atribuição de tarefas

Cada membro contribuiu com uma parte de montar o sistema, aqui está uma tabela com o que cada um fez.

<b>Nathan</b>	<b>Gabriel de Almeida</b>	<b>Enzo</b>	<b>Enzo Alberoni</b>	<b>Mariana</b>
Atividades	Atividades	UML	Cronograma	UML
Login	Histórico	Cronograma	UML	Relatório
Documento	Requisição	Atividades	Design	Design
UML	UML	Relatório	Relatorio	Histórico

Tabela 3.1: Tabela de divisão de tarefas







## 4. Resultados

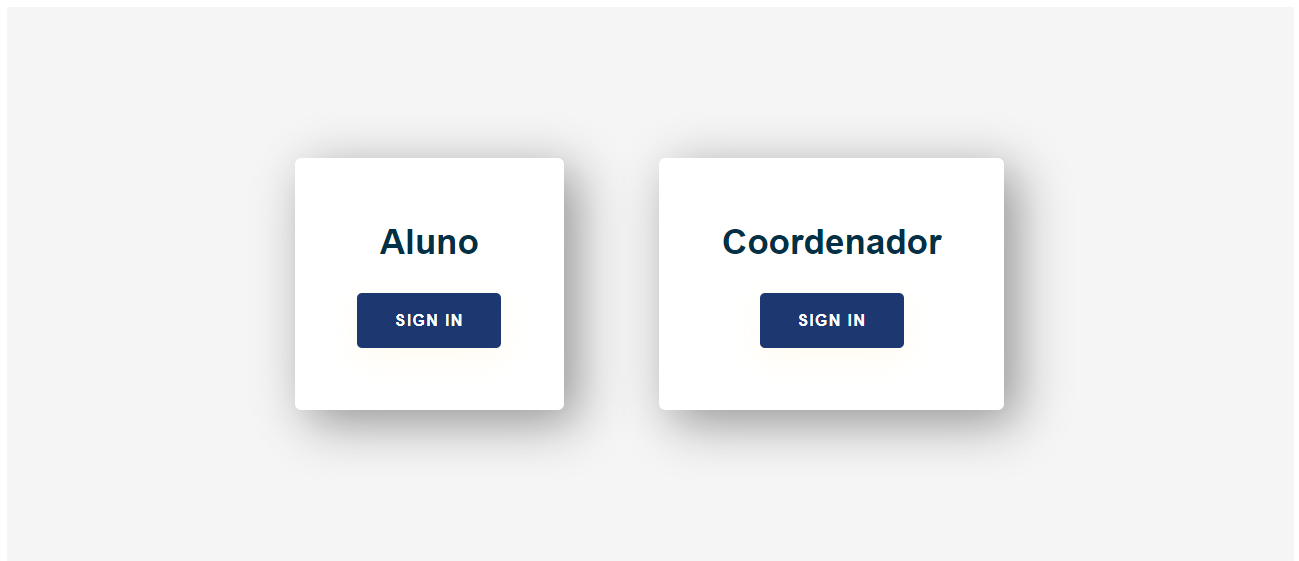
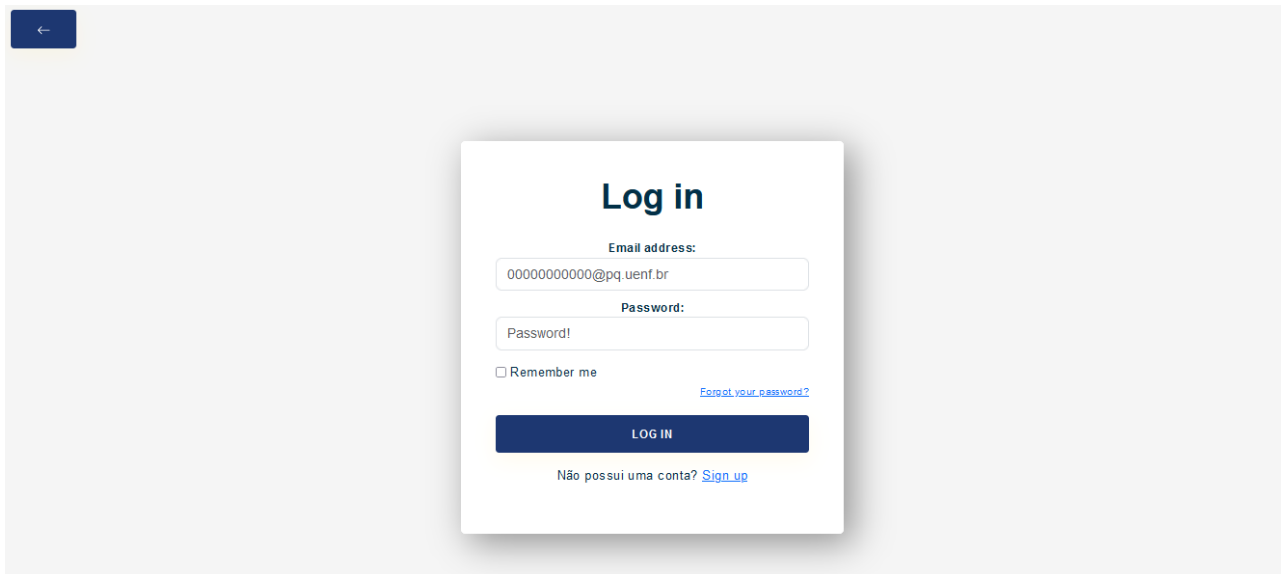


Figura 4.1: Página Inicial

Essa é página inicial do sistema. Onde o usuário tem capacidade de entrar como aluno ou coordenador.

A login form titled "Log in" is centered on a light gray background. It features a dark blue back arrow button in the top left. The form has two input fields: "Email address:" with the value "00000000000@pq.uenf.br" and "Password:" with the placeholder "Password!". Below the password field is a "Remember me" checkbox and a "Forgot your password?" link. A dark blue "LOG IN" button is at the bottom of the form. Below the button is a link "Não possui uma conta? Sign up".

**Log in**

Email address:  
00000000000@pq.uenf.br

Password:  
Password!

☐ Remember me [Forgot your password?](#)

**LOG IN**

Não possui uma conta? [Sign up](#)

Figura 4.2: Login

Esse é nosso sistema de login onde coloca seu email e senha e entra.

The home page has a dark blue header with "HOME" on the left and a user profile icon on the right. The main content area has a light gray background. It features two sections: "ACE I – Projetos e Programas" and "ACE II – Cursos e Eventos". Each section contains a table with columns: Titulo, Orientador, Horas Feitas, Status, and Saiba mais. The "ACE I" table has 3 data rows and a total row. The "ACE II" table has 4 data rows and a total row.

**HOME**

**ACE I – Projetos e Programas**

Título:	Orientador:	Horas Feitas:	Status:	Saiba mais:
segunda feira	Nestoraas	18	validated	<a href="#">saiba mais</a>
guia latex for babies	ausberto	36	validated	<a href="#">saiba mais</a>
Esse aqui tem doc	asdqweqsd	45	rejected	<a href="#">saiba mais</a>
Total de Horas:			54.0	

**ACE II – Cursos e Eventos**

Título:	Orientador:	Horas Feitas:	Status:	Saiba mais:
segunda feira		17	validated	<a href="#">saiba mais</a>
testando23123		15	validated	<a href="#">saiba mais</a>
Quiche de frango		23	validated	<a href="#">saiba mais</a>
AAAAA		40	rejected	<a href="#">saiba mais</a>
Total de Horas:			55.0	

Figura 4.3: Home

A página inicial do aluno onde mostra todas as atividades que ele realizou e permite cadastrar novas atividades.

**Nova atividade**

Título

Ruby on rails

Ace

ACE I – Projetos e Programas

Orientador

Annabell

Horas feitas

30

Documento

Procurar... 125918090-1.pdf

**CADASTRAR ATIVIDADE**

Figura 4.4: CadastroAtividades

A aba de cadastrar atividades onde o aluno informa as informações da atividade solicita o cadastro

**ACE I – Projetos e Programas**

Título:	Orientador:	Horas Feitas:	Status:	Saiba mais:
segunda feira	Nestoras	18	validated	<a href="#">saiba mais</a>
guia latex for babies	ausberto	36	validated	<a href="#">saiba mais</a>
Esse aqui tem doc	asdqwesd	45	rejected	<a href="#">saiba mais</a>
Ruby on rails	Annabell	30	pending	<a href="#">saiba mais</a>
Total de Horas:			54.0	

Figura 4.5: Atividade Pendente

Atividade exibe para o aluno e fica com status 'pendentes'. As horas só são válidas se tiver status 'Validated'.



Título:	Ace:	Horas Feitas:	Email:	Status:	Validar:
Alberoni	ACE III – Prestação de Serviços	15.0	gv.almeida06@gmail.com	pending	<a href="#">saiba mais</a>
Estudar	ACE II – Cursos e Eventos	15.0	mariana@gmail.com	pending	<a href="#">saiba mais</a>
Almaida Fake	ACE III – Prestação de Serviços	97.0	nathan@example.com	pending	<a href="#">saiba mais</a>
Ruby on rails	ACE I – Projetos e Programas	30.0	gv.almeida06@gmail.com	pending	<a href="#">saiba mais</a>

Figura 4.6: Coordenador

Essa é a aba do coordenador onde ele vê a lista de todas as atividades pendentes e suas respectivas informações.

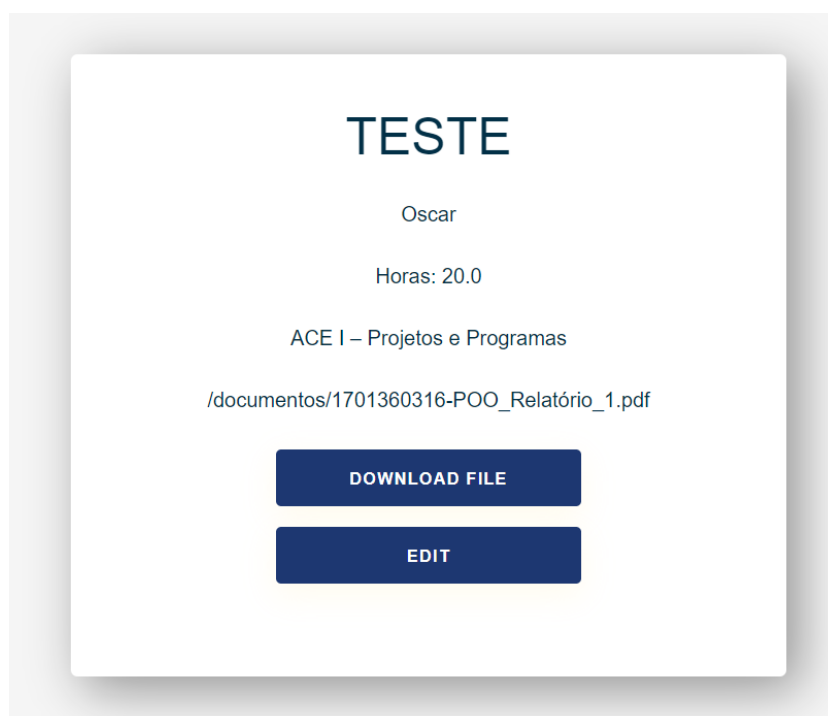


Figura 4.7: Pegar Documento

O coordenador tem a opção de pegar o documento enviado que comprava que a atividade foi feita. Com isso, confere os dados e decide se valida ou não

## Editar atividade

Titulo

TESTE

Ace

ACE I – Projetos e Programas

Orientador

Nathan

Horasfeitas

20

Status

rejected

ATUALIZAR ATIVIDADE

Figura 4.8: Validar

Aqui o coordenador decide se ele valida ou rejeita a atividade.

[Home](#) [Atividades](#)



### Atividades Pendentes

Titulo:	Ace:	Horas Feitas:	Email:	Status:	Validar:
Alberoni	ACE III – Prestação de Serviços	15.0	gv.almeida06@gmail.com	pending	<a href="#">saiba mais</a>
Estudar	ACE II – Cursos e Eventos	15.0	mariana@gmail.com	pending	<a href="#">saiba mais</a>
Almeida Fake	ACE III – Prestação de Serviços	97.0	nathan@example.com	pending	<a href="#">saiba mais</a>

Figura 4.9: Home Coordenador

Apos validar, a atividade desaparece da home inicial do coordenador.

## Histórico

Título	ACE	Coordenador	Horas Feitas	Status
Aparece aí?	ACE IV – Disciplinas Extensionista		2.0	validated
Aparece aí?	ACE IV – Disciplinas Extensionista		10.0	validated
segunda feira	ACE I – Projetos e Programas	Nestoraas	18.0	validated
segunda feira	ACE II – Cursos e Eventos		17.0	validated
prova do tang	ACE III – Prestação de Serviços		28.0	rejected
FINALMENTE	ACE III – Prestação de Serviços		15.0	validated
testeDiogo	ACE III – Prestação de Serviços		10.0	validated
testando23123	ACE II – Cursos e Eventos		15.0	validated
SCTI	ACE II – Cursos e Eventos	tang	15.0	validated
Relógio	ACE II – Cursos e Eventos		45.0	validated
Cantar	ACE I – Projetos e Programas	Oscar	35.0	validated
Quiche de frango	ACE II – Cursos e Eventos		23.0	validated
terca tango	ACE III – Prestação de Serviços		10.0	validated
guia latex for babies	ACE I – Projetos e Programas	ausberto	36.0	validated
Calebrito	ACE IV – Disciplinas Extensionista		20.0	validated
Neitan	ACE III – Prestação de Serviços		50.0	validated

Figura 4.10: Histórico

Aqui é a aba de histórico onde mostra as atividades de cada aluno e suas respectivas informações.

## ACE I – Projetos e Programas

Título:	Orientador:	Horas Feitas:	Status:	Saiba mais:
segunda feira	Nestoraas	18	validated	<a href="#">saiba mais</a>
guia latex for babies	ausberto	36	validated	<a href="#">saiba mais</a>
Esse aqui tem doc	asdqwesd	45	rejected	<a href="#">saiba mais</a>
Ruby on rails	Annabell	30	validated	<a href="#">saiba mais</a>
Total de Horas:			84.0	

Figura 4.11: Atividade Validada

Por fim, ao retornar na aba de alunos, a atividade foi validada e sua hora contabilizada.



