

Projeto sigAda

Eduardo Barbosa

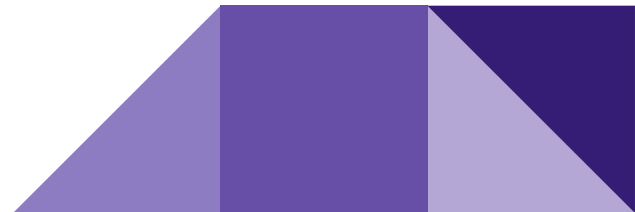
Rafael Cardoso

Ricardo Bayer

Tainá Lima

Apresentação

- O sigAda
- Metodologia
- Documentação
- Implementação
- Dificuldades e Soluções
- Software em funcionamento



Sobre o projeto

O sigAda

- O que é
- Funcionalidades principais
- Público-alvo
- Ambiente de execução



MINERVA SOLUTIONS

< SIG *ada* >

I N S T I T U T O
< *ada lovelace* >
D E E N S I N O

Metodologia

O scrum

- O que é?
- Por que escolhemos scrum?
- Desvantagens

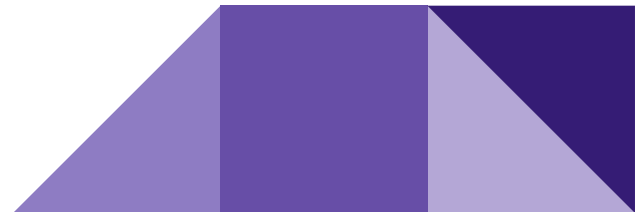


Documentação

O que é, como foi utilizado e dificuldades








Plano de Projetos

- Gerenciamento de Tempo
 - Divisão das sprints
 - Quadros do pipefy
 - Organização interna das sprints
 - Burndown



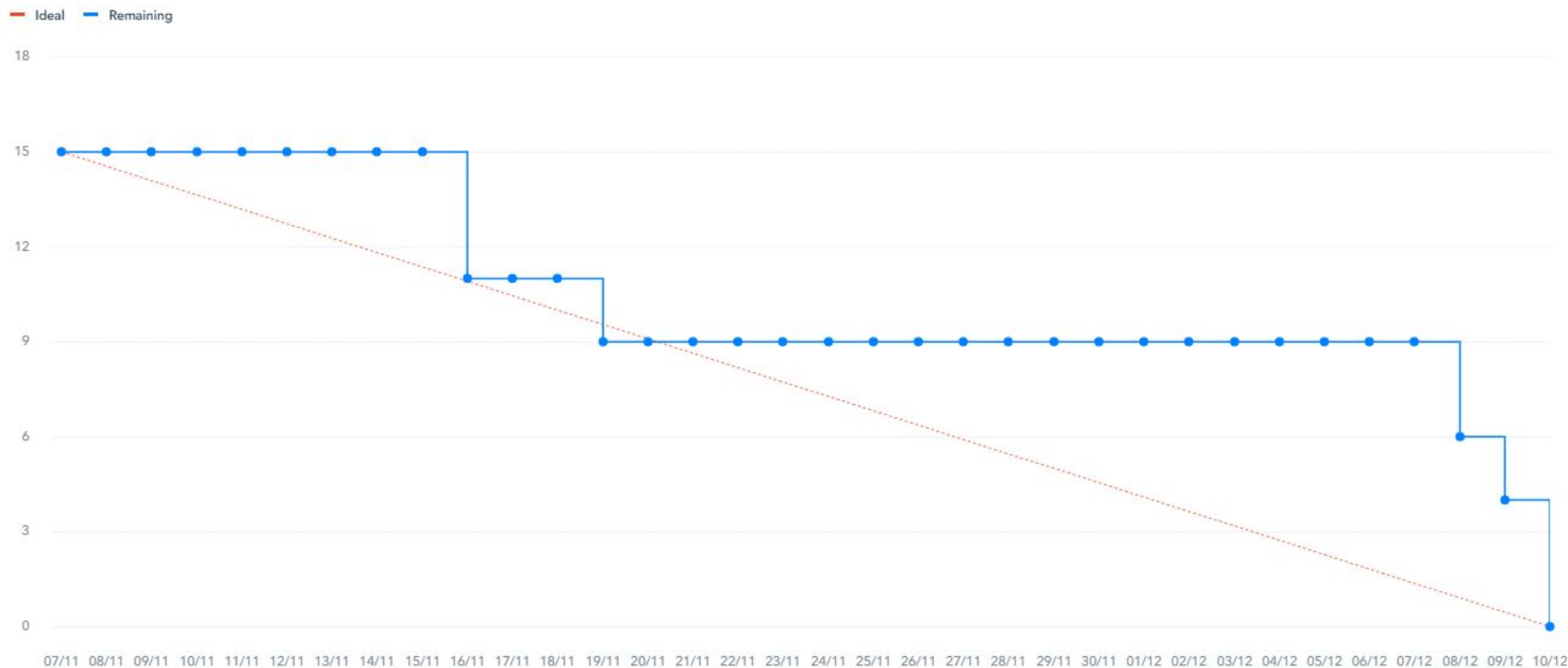
Plano de Projetos

 Sprint 1 - Marco 2  0%

Documentos de requisitos n... 0% 	Backlog de Produto (0)	Sprint (0)	Fazendo (0)
Planejamento das iterações (... 0% 	Backlog de Produto (0)	Sprint (0)	Fazendo (0)
Diagrama de sequencia (Mar... 0% 	Backlog de Produto (0)	Sprint (0)	Fazendo (0)
Documento de arquitetura (... 0% 	Backlog de Produto (0)	Sprint (0)	Fazendo (0)
Gráfico burndown (Marco 2) 0% 	Backlog de Produto (0)	Sprint (0)	Fazendo (0)
Diagrama de classe (Marco 2) 0% 	Backlog de Produto (0)	Sprint (0)	Fazendo (0)
Diagrama de atividade (Mar... 0% 	Backlog de Produto (0)	Sprint (0)	Fazendo (0)






Plano de Projetos

Sprint 3 - Marco 3 (7 de nov de 2018 - 10 de dez de 2018)



Plano de Projetos

- Gerenciamento de Comunicação e Projeto

#	Logo	Ferramenta	Descrição
1		Telegram	Utilizado para mensagens rápidas, fixar mensagens importantes, guardar links importantes e marcar reuniões.
2		Drive	Utilizado para guardar a documentação, planilhas, tabelas, resumos, relatórios, entre outros.
3		Discord	Utilizado para fazer reuniões da sprint, fixar mensagens importantes, guardar links importantes e desenvolvimento colaborativo em tempo real.
4		GitHub	Utilizado para armazenar os códigos desenvolvidos.
5		Pipefy	Utilizado para gerenciamento do projeto. Usamos o Kanban em conjunto com o Scrum

Plano de Projetos

- Gerenciamento de riscos

#	Risco	Probabilidade	Impacto	Exposição	Plano de Resposta
1	Inviabilidade de uma pessoa chave	Moderada	Grave	Atraso do cronograma	Dependendo do tempo e motivos dessa inviabilidade e a carga de trabalho que este possuía, contratar outra pessoa ou remanejar suas tarefas para outros funcionários
2	Falha de energia	Baixo	Moderado	Atraso do cronograma	Continuar o desenvolvimento do projeto remotamente

Plano de Projetos

- Gerenciamento de riscos

6	Dessincronizaçã o dos times	Alta	Moderado	Coloca em risco a qualidade do sistema	Realocação dos times
▼					
7	Perda de conexão da internet	Alta	Grave	Impossibilita o acesso a dados/informações remotos e comunicação entre as partes interessadas por meio da internet	Contratação de plano de internet auxiliar.

Documento de Visão

- Usuários
- Divisão de trabalho
- Principais funcionalidades
- Requisitos funcionais
- Requisitos não-funcionais
 - Categoria
 - Prioridade
 -

Identificador	RNF.DES.003	Categoria	Desempenho
Nome	O sistema deve ter uma tempo de execução inferior 30 segundos		
Data de criação	15/11/2018	Autor	Rafael, Tainá e Eduardo
Data da última alteração	15/11/2018	Autor	Rafael, Tainá e Eduardo
Versão	1.0	Prioridade	Desejável
Descrição	O tempo de execução pequeno deve ser levado em consideração, uma vez que torna o uso do sistema pelo usuário mais fluido, além não fazer com que haja pouca aderência ao sistema.		

Diagrama de casos de uso

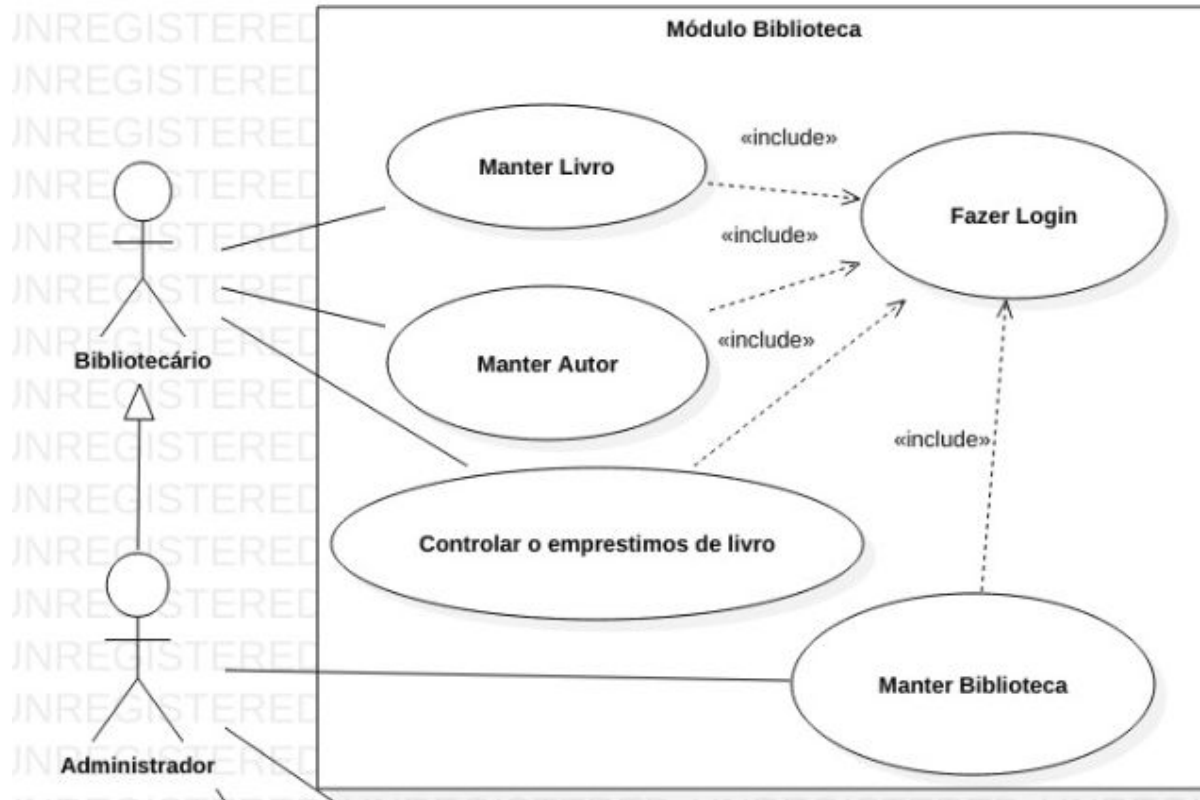


Diagrama de casos de uso

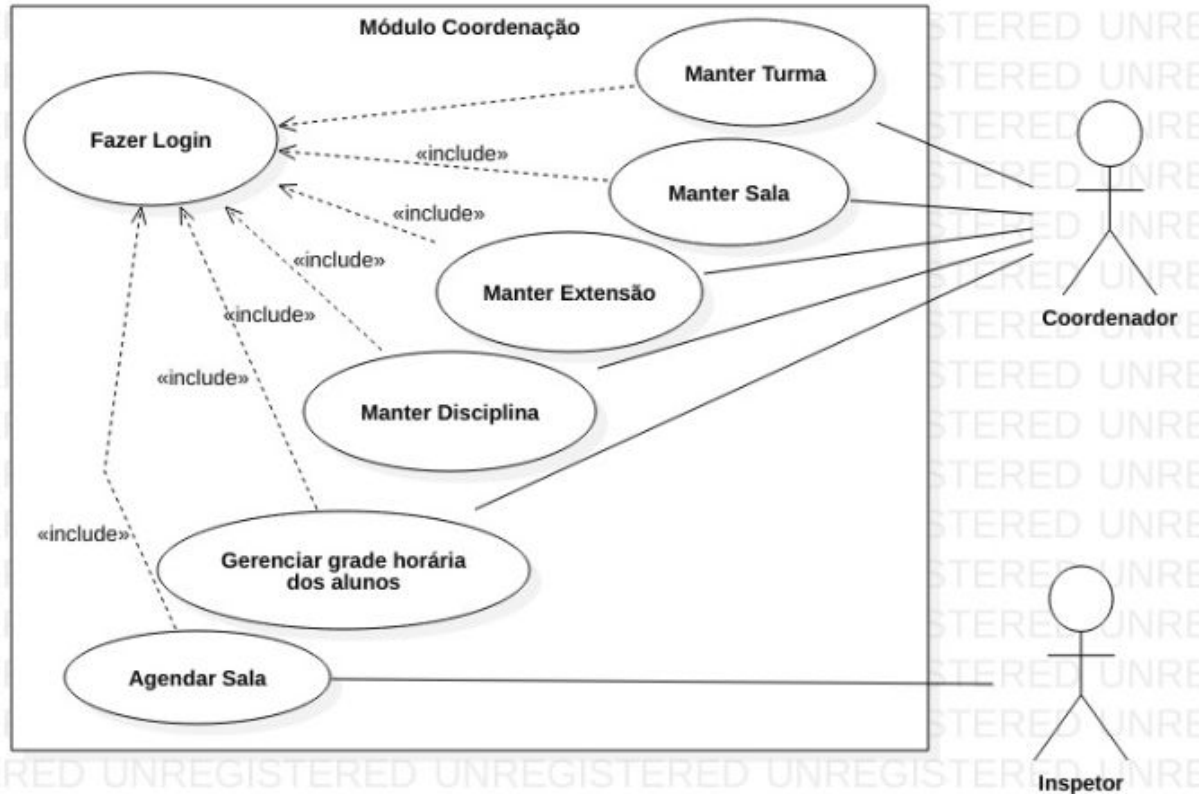
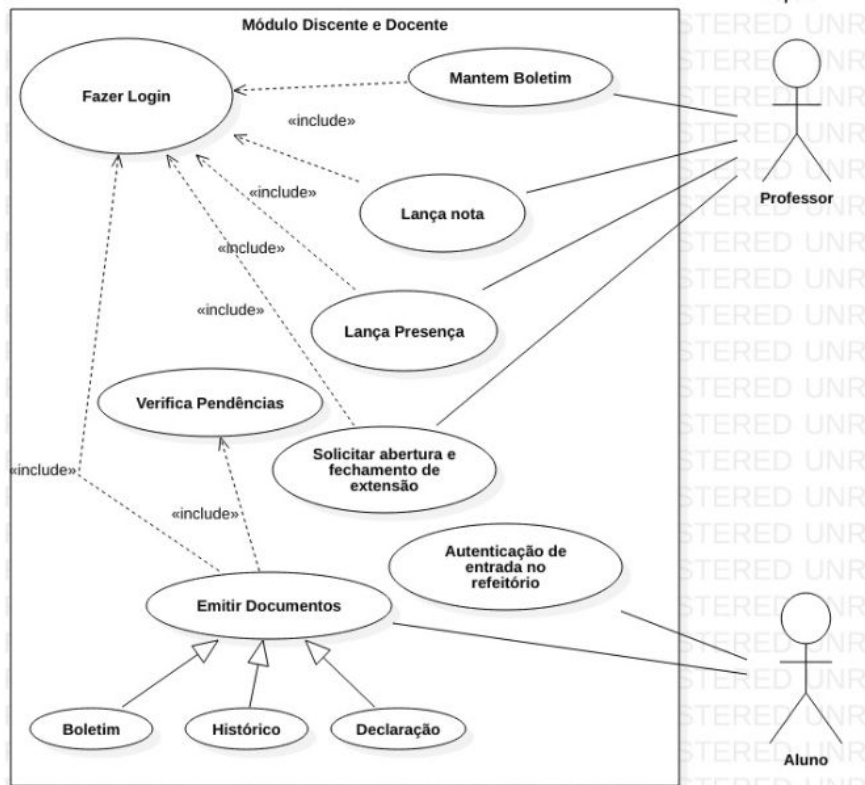
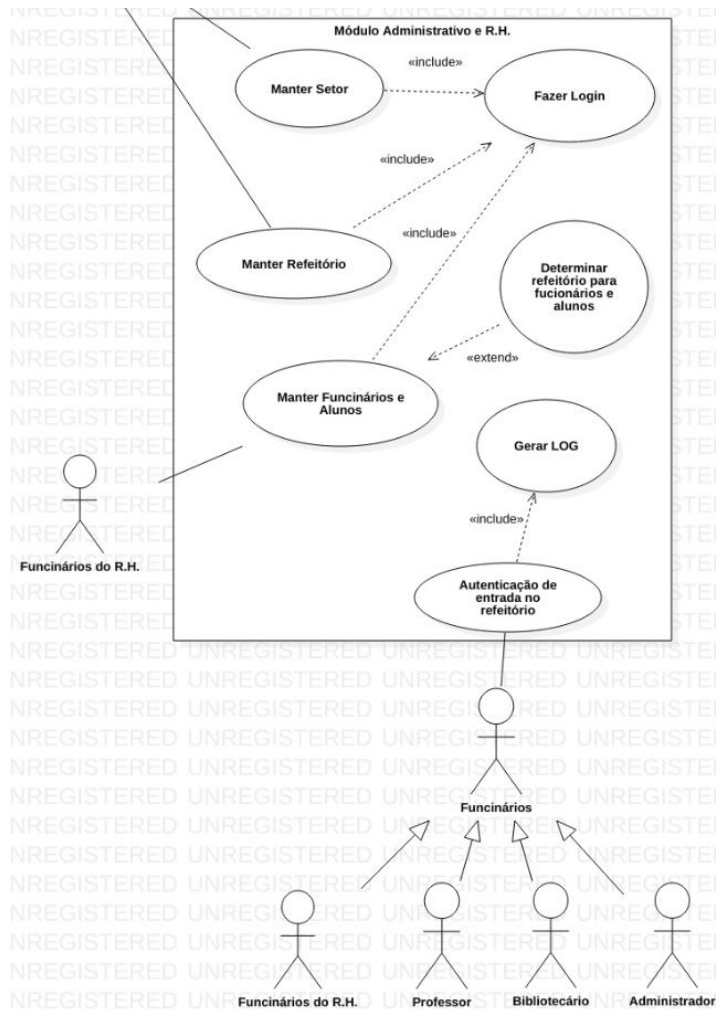



Diagrama de casos de uso



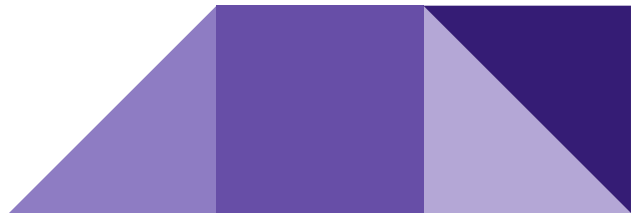


Especificação dos casos de uso

- CS01 Manter Aluno/Funcionário
 - Descrição
 - Ator(es)
 - Pré-condições
 - Fluxo principal
 - Regras de negócio
- 

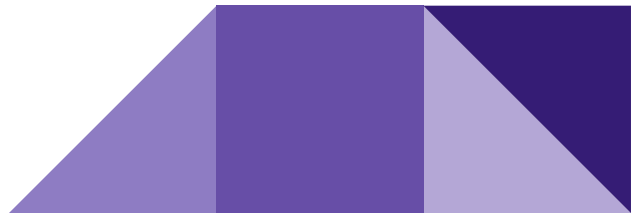
Planejamento das Iterações

- As iterações: quantidade e o que foi feito
- Organização das iterações
 - Atividades da iteração
 - Recursos humanos e tecnológicos
 - Casos de uso
 - Critério de avaliação e planos para a próxima iteração



Planejamento das Iterações

- Sobre o caso de uso CS01 Manter Aluno/Funcionário
 - Iteração de criação e especificação
 - Iteração de atualização de informações sobre este caso de uso
 - Iteração da implementação deste caso de uso



```

classDiagram
    class User {
        id int
        name String
        email String
        phone String
        password String
        role String
    }
    class Admin {
        id int
        name String
        email String
        phone String
        password String
        role String
    }
    class Manager {
        id int
        name String
        email String
        phone String
        password String
        role String
    }
    class Employee {
        id int
        name String
        email String
        phone String
        password String
        role String
    }
    class Customer {
        id int
        name String
        email String
        phone String
        password String
        role String
    }
    class Supplier {
        id int
        name String
        email String
        phone String
        password String
        role String
    }
    class Dealer {
        id int
        name String
        email String
        phone String
        password String
        role String
    }
    class Distributor {
        id int
        name String
        email String
        phone String
        password String
        role String
    }
    class Manufacturer {
        id int
        name String
        email String
        phone String
        password String
        role String
    }

    User "1" -- "1" Admin : generalization
    User "1" -- "1" Manager : generalization
    User "1" -- "1" Employee : generalization
    User "1" -- "1" Customer : generalization
    User "1" -- "1" Supplier : generalization
    User "1" -- "1" Dealer : generalization
    User "1" -- "1" Distributor : generalization
    User "1" -- "1" Manufacturer : generalization

    User "1" -- "1" Admin : association
    User "1" -- "1" Manager : association
    User "1" -- "1" Employee : association
    User "1" -- "1" Customer : association
    User "1" -- "1" Supplier : association
    User "1" -- "1" Dealer : association
    User "1" -- "1" Distributor : association
    User "1" -- "1" Manufacturer : association

    Admin "1" -- "1" Manager : association
    Admin "1" -- "1" Employee : association
    Admin "1" -- "1" Customer : association
    Admin "1" -- "1" Supplier : association
    Admin "1" -- "1" Dealer : association
    Admin "1" -- "1" Distributor : association
    Admin "1" -- "1" Manufacturer : association

    Manager "1" -- "1" Employee : association
    Manager "1" -- "1" Customer : association
    Manager "1" -- "1" Supplier : association
    Manager "1" -- "1" Dealer : association
    Manager "1" -- "1" Distributor : association
    Manager "1" -- "1" Manufacturer : association

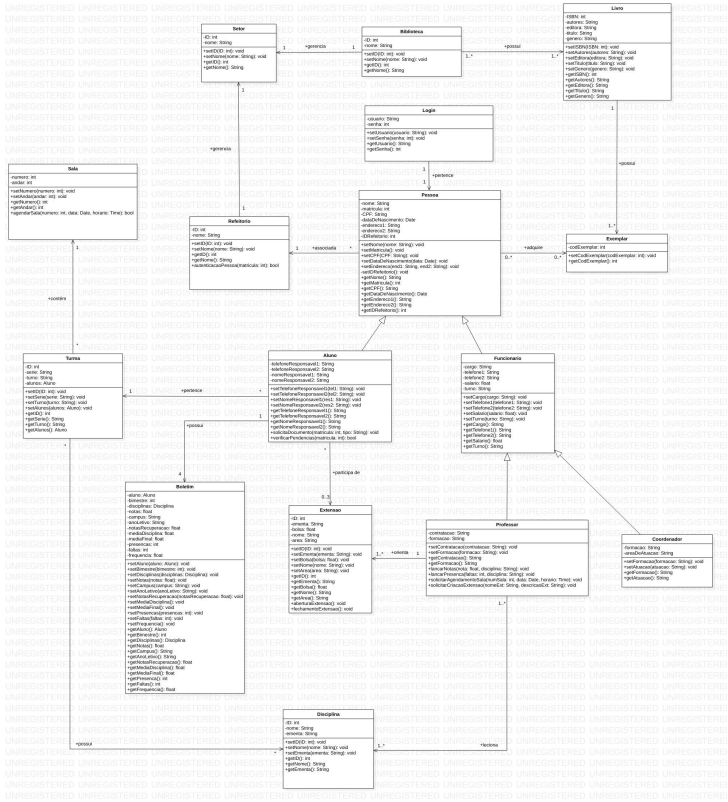
    Employee "1" -- "1" Customer : association
    Employee "1" -- "1" Supplier : association
    Employee "1" -- "1" Dealer : association
    Employee "1" -- "1" Distributor : association
    Employee "1" -- "1" Manufacturer : association

    Customer "1" -- "1" Supplier : association
    Customer "1" -- "1" Dealer : association
    Customer "1" -- "1" Distributor : association
    Customer "1" -- "1" Manufacturer : association

    Supplier "1" -- "1" Dealer : association
    Supplier "1" -- "1" Distributor : association
    Supplier "1" -- "1" Manufacturer : association

    Dealer "1" -- "1" Distributor : association
    Dealer "1" -- "1" Manufacturer : association

    Distributor "1" -- "1" Manufacturer : association
  
```



Diagrama

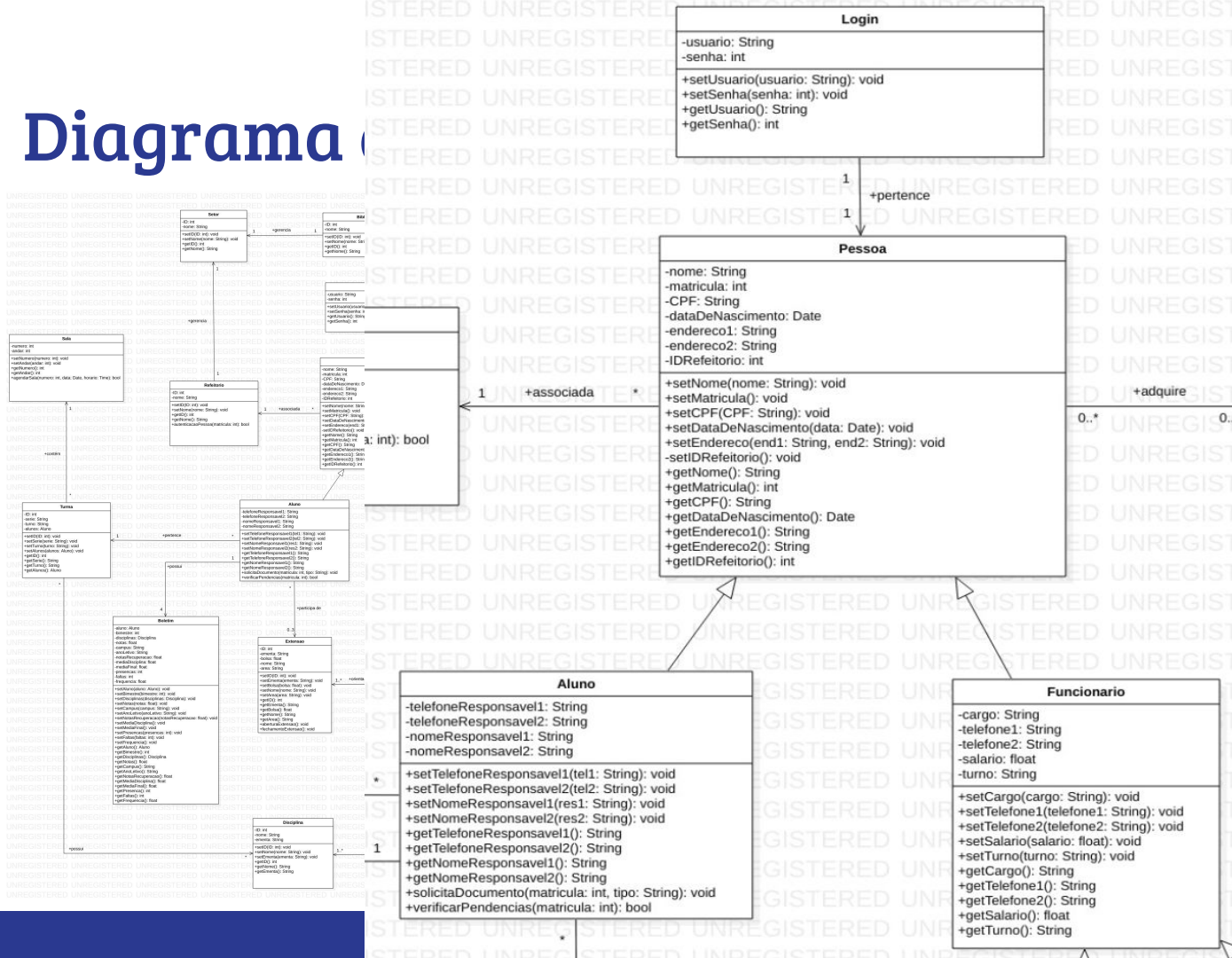


Diagrama de sequência

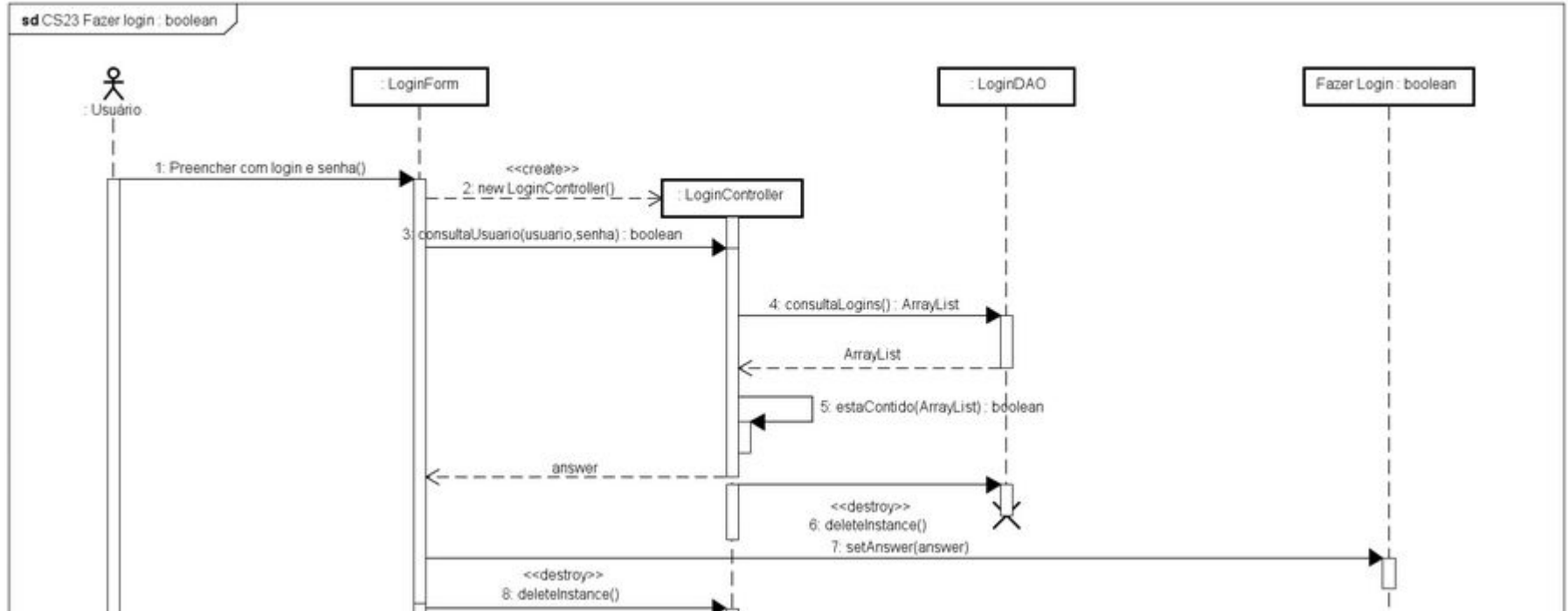


Diagrama de sequência

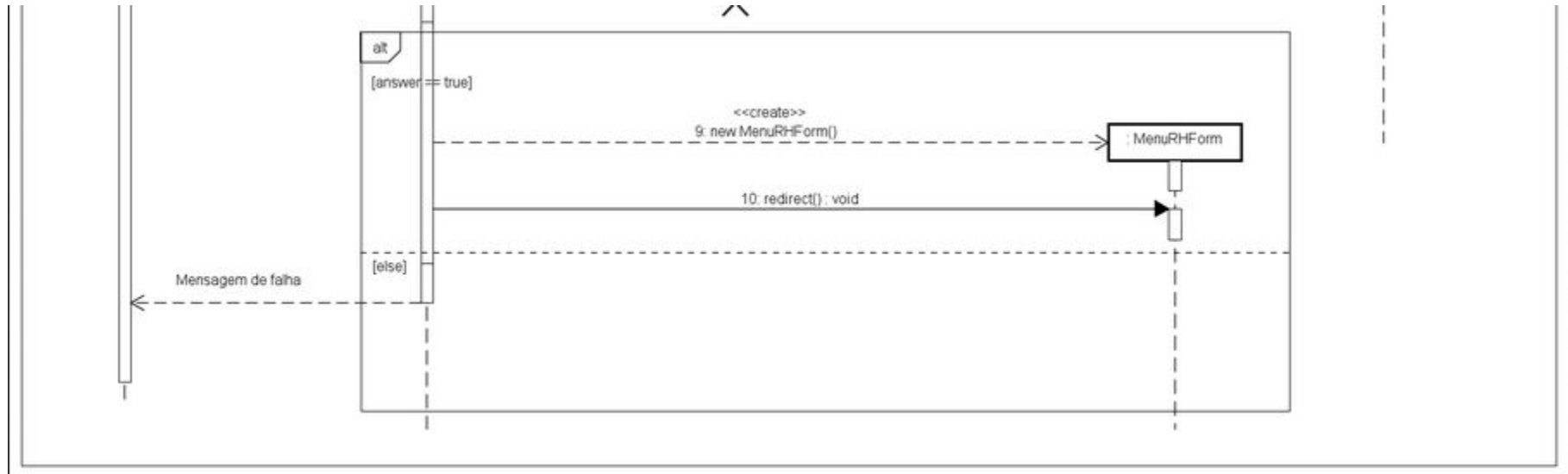
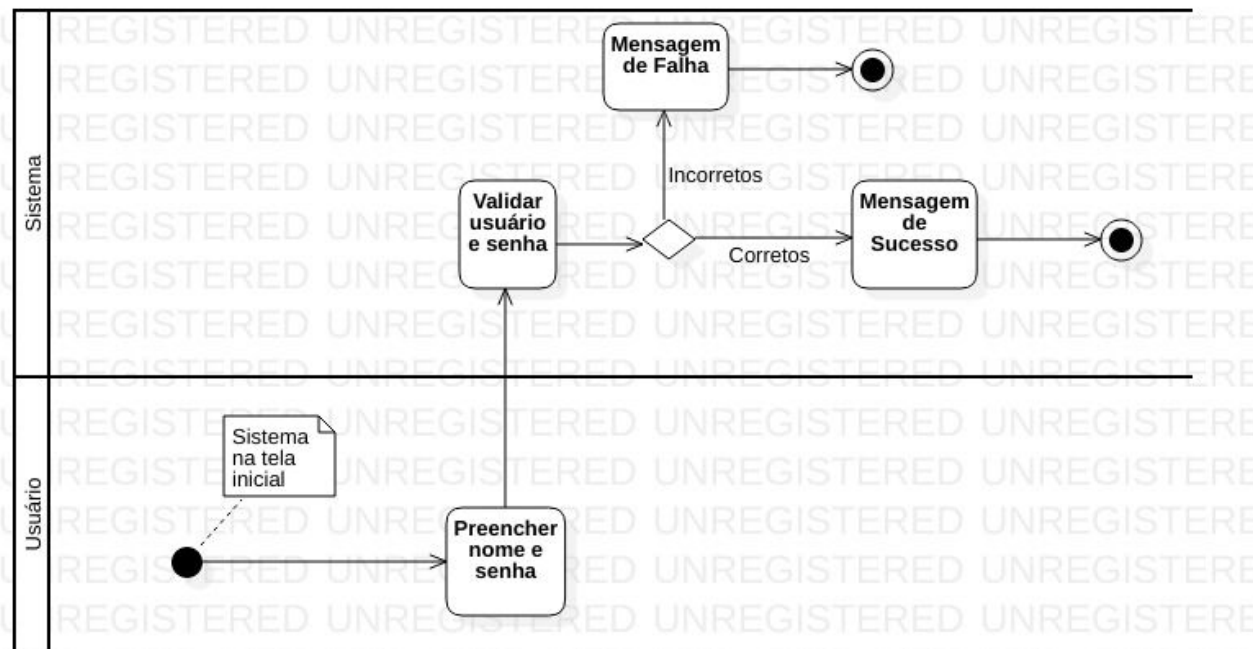


Diagrama de atividade



Documento de arquitetura

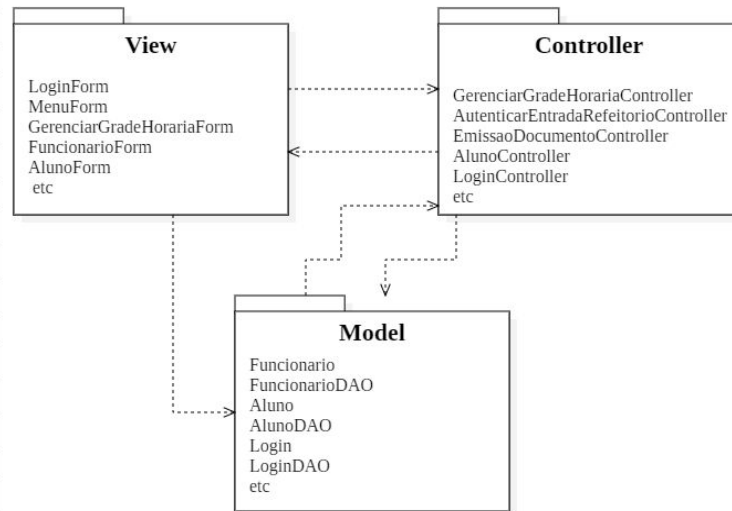
1. Metas e restrições da arquitetura

2. Visão Lógica

3. Padrão de codificação

- Declaração de pacotes
- Declaração de classes/ interfaces
- Variáveis constantes
- Declaração de métodos
- Variáveis

Diagrama de Pacotes



Documento de arquitetura

```
package com.ms.sigada.controller;

import java.util.Map;

public class AlunoController {
    private Map<Integer, Aluno> alunos = new TreeMap<>();

    public AlunoController() {
        alunos = AlunoDAO.consultaAlunos();
    }

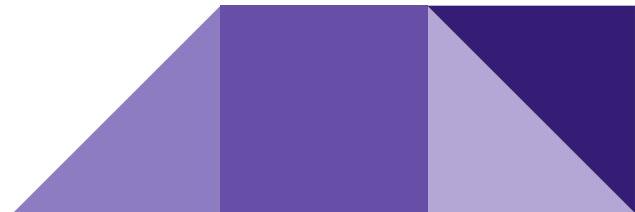
    public boolean cadastraAluno(int matricula, String nome, String dataNasc, String nomeRes1,
        String nomeRes2, String end1, String end2, String tel1, String tel2, String cpf){
        Aluno novoAluno = new Aluno();
        int maiorMatricula = 0;
        if(matricula == 0) {
            for(int matriculaAluno : alunos.keySet()) {
                if(matriculaAluno > maiorMatricula) {
                    maiorMatricula = matriculaAluno;
                }
            }
            novoAluno.setMatricula(maiorMatricula+1);
        } else {
            novoAluno.setMatricula(matricula);
        }

        novoAluno.setIDRefeitorio(1);
        novoAluno.setNome(nome);
        novoAluno.setCPF(cpf);
        novoAluno.setDataDeNascimento(dataNasc);
        novoAluno.setEndereco1(end1);
        novoAluno.setEndereco2(end2);
        novoAluno.setTelefoneResponsavel1(tel1);
        novoAluno.setTelefoneResponsavel2(tel2);
        novoAluno.setNomeResponsavel1(nomeRes1);
        novoAluno.setNomeResponsavel2(nomeRes2);
    }
}
```

Plano de gerenciamento configuração

Tipo	Ferramenta	Versão	Função
Controle de Versão	Git e GitHub	-	Repositório para os códigos e documentação e o mesmo será responsável pelo versionamento
Controle de Versão	Google Drive	-	Repositório da documentação e o mesmo será responsável pelo versionamento
IDE	Eclipse e/ou Netbeans	-	Utilizado para a implementação do software, tendo integração com o GitHub, para que se possa atualizar o código armazenado nos repositórios
Editor de Texto	Google Docs	-	Utilizado para a criação dos artefatos gerados no projeto
















- Ferramentas, ambiente e infraestrutura



Plano de gerenciamento configuração

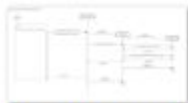
- Documentação
 - <TIPO_ART>_InstitutoAdaLovelace.<TIPO_EXT>
- Imagem
 - <SIGLA_TIPO_IMG>_<DESCRIÇÃO_IMG>_InstitutoAdaLovelace.<TIPO_EXT>
- Versionamento nos Docs
 - <primeiro>.<segundo>.<terceiro>
- Comentários de release no GitHub
 - <SIMBOLO><COMENTARIO>

Plano de gerenciamento configuração

	PlanoDeGerenciamentoDeConfiguracao_InstitutoAdaLov...	 me	8:44 PM	
	PlanoDeControleDeQualidade_InstitutoAdaLovelace	 me	8:33 PM	
	DocumentoDeArquitetura_InstitutoAdaLovelace	 R. Cardoso Cardoso	6:53 PM	
	DocumentoDeVisao_InstitutoAdaLovelace	 R. Cardoso Cardoso	5:18 PM	
	PlanoDeIteracoes_InstitutoAdaLovelace	 me	4:41 PM	



DS_CS01ManterA
lunov4_Instituto
AdaLovelace



DS_CS09DeterminarRefeitorioPara
AlunoFuncv2_Ins
titutoAdaLovel...



DS_CS10AutenticacaoDeEntradaN
oRefv2_Instituto
AdaLovelace



DS_CS11Gerencia
rGradeHorariaAlu
nosv2_InstitutoA
daLovelace



DS_CS23
FazerLoginv2_Ins
titutoAdaLovelac
e

Plano de gerenciamento configuração

📁 .settings	+ Arquivos default do Projeto em Java
📁 Documentos	# Atualização e correção parcial dos documentos; + RelatorioDeResulta...
📁 bin	+Testes Junit; #AlunoController; #ManterFormAluno; #AlunoDAO
📁 sigAdaForms	#Correcocoes nos arquivos; +Funcao deletar
📁 src	+Testes Junit; #AlunoController; #ManterFormAluno; #AlunoDAO
📄 .classpath	+ Arquivos default do Projeto em Java
📄 .project	+ Arquivos default do Projeto em Java
📄 Ada.jar	+Testes Junit; #AlunoController; #ManterFormAluno; #AlunoDAO

Plano de gerenciamento configuração

Data	Versão	Descrição	Autor(es)
22/09/2018	1.0.0	Abertura do Documento	Eduardo, Rafael , Tainá e Ricardo
07/11/2018	1.1.0	Inclusão da tabela de histórico de versão e atualização do diagrama de caso de uso	Eduardo
15/11/2018	1.2.0	Alteração do caso de uso CS11, atualização do diagrama de caso de uso e criação do caso de uso CS23	Eduardo, Tainá, Rafael e Ricardo
16/11/2018	1.3.0	Adição do CS “Verifica Pendências” e atualização do diagrama.	Eduardo
08/12/2018	2.0.0	Correção das regras de negócio de todos os casos de uso.	Rafael e Eduardo

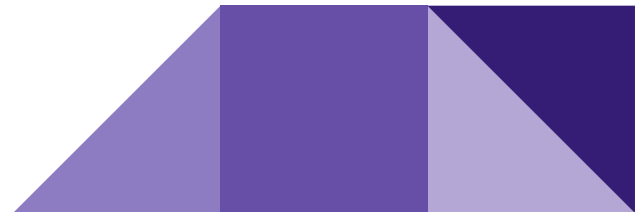
Plano de gerenciamento configuração

- Check-out e check-in
- Estratégia de branch
- Estratégia de merge

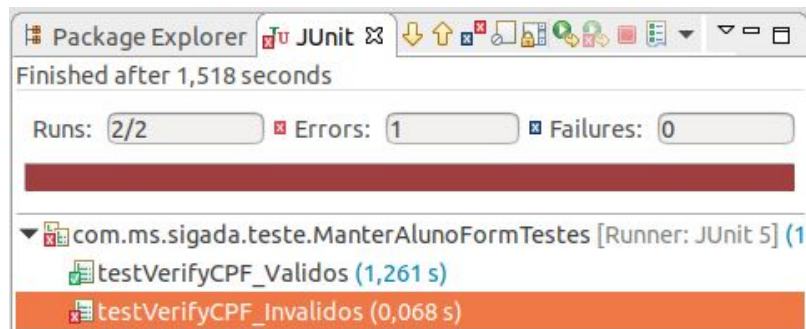
Default branch	
master	Updated 22 hours ago by Eduardodsb
Active branches	
HotfixManterAlunoFormRafael	Updated 2 days ago by Cardosorf
ViewEduardo	Updated 22 days ago by Eduardodsb
DAOEduardo	Updated 22 days ago by Eduardodsb
HotfixDAORafael	Updated 22 days ago by Cardosorf
DAORafael	Updated 22 days ago by Cardosorf

Plano de testes e inspeção

- Teste de unidade
- Teste de integração
- Teste de sistema
- Análise estática automatizada
 - Má prática
 - Corretude
 - Evasivo



Plano de testes e inspeção



```
package com.ms.sigada.teste;
```

```
import static org.junit.Assert.*;
```

```
public class ManterAlunoFormTestes {
```

```
    @Test
```












```
    public final void testVerifyCPF_Validos() {  
        ManterAlunoForm t = new ManterAlunoForm("");  
        Assert.assertEquals(true, t.verifyCPF("47763581077"));  
        Assert.assertEquals(true, t.verifyCPF("49870235123"));  
        Assert.assertEquals(true, t.verifyCPF("05093936081"));  
        Assert.assertEquals(true, t.verifyCPF("13372061055"));  
    }
```

```
    @Test
```

```
    public final void testVerifyCPF_Invalidos() {  
        ManterAlunoForm t = new ManterAlunoForm("");  
        Assert.assertEquals(false, t.verifyCPF("11111111111"));  
        Assert.assertEquals(false, t.verifyCPF("87415447545"));  
        Assert.assertEquals(false, t.verifyCPF("65485475422"));  
        Assert.assertEquals(false, t.verifyCPF("89754124586"));  
        Assert.assertEquals(false, t.verifyCPF("34651238569"));  
        Assert.assertEquals(false, t.verifyCPF("abcdefrthyu"));  
    }
```

```
}
```

Plano de testes e inspeção

- ✓  > Projeto_InstitutoAdaLovelace (4) [Projeto_InstitutoAdaLovelace master]
 - ✓  Troubling (1)
 - ✓  Normal confidence (1)
 - ✓  Comparison of String objects using == or != (1)
 -  Comparison of String objects using == or != in com.ms.sigada.view.ManterAlunoForm\$5.actionPerformed(ActionEvent) [Troubling(11), Normal confidence]
 - ✓  Of Concern (3)
 - ✓  High confidence (3)
 - ✓  Write to static field from instance method (3)
 -  Write to static field com.ms.sigada.view.ManterAlunoForm.file1 from instance method new com.ms.sigada.view.ManterAlunoForm(String) [Of Concern(15), High confidence]
 -  Write to static field com.ms.sigada.view.ManterAlunoForm.path from instance method new com.ms.sigada.view.ManterAlunoForm(String) [Of Concern(15), High confidence]
 -  Write to static field com.ms.sigada.view.MenuRHForm.file1 from instance method new com.ms.sigada.view.MenuRHForm(String) [Of Concern(15), High confidence]

Plano de testes e inspeção

Bug: Comparison of String objects using == or != in com.ms.sigada.view.ManterAlunoForm\$5.actionPerformed(ActionEvent)

This code compares java.lang.String objects for reference equality using the == or != operators. Unless both strings are either constants in a source file, or have been interned using the String.intern() method, the same string value may be represented by two different String objects. Consider using the equals(Object) method instead.

Rank: Troubling (11), **confidence:** Normal

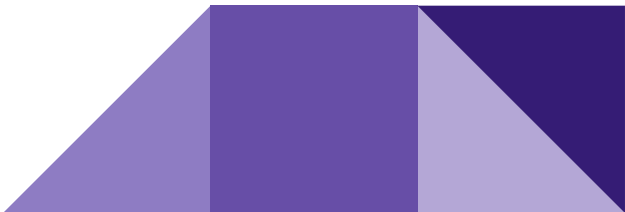
Pattern: ES_COMPARING_STRINGS_WITH_EQ

Type: ES, **Category:** BAD_PRACTICE (Bad practice)

XML output:

```
<BugInstance type="ES_COMPARING_STRINGS_WITH_EQ" priority="2" rank="11" abbrev="ES" category="BAD_PRACTICE" first="1">
  <Class classname="com.ms.sigada.view.ManterAlunoForm$5">
    <SourceLine classname="com.ms.sigada.view.ManterAlunoForm$5" sourcefile="ManterAlunoForm.java" sourcepath="com/ms/sigada/view/Mant
  </Class>
  <Method classname="com.ms.sigada.view.ManterAlunoForm$5" name="actionPerformed" signature="(Ljava/awt/event/ActionEvent;)V" isStatic=
    <SourceLine classname="com.ms.sigada.view.ManterAlunoForm$5" start="307" end="334" startBytecode="0" endBytecode="94" sourcefile="M
  </Method>
  <Type descriptor="Ljava/lang/String;" role="TYPE_FOUND">
    <SourceLine classname="java.lang.String"/>
  </Type>
  <String value="" role="STRING_CONSTANT"/>
  <LocalVariable name="matriculaTemp" register="4" pc="27" role="LOCAL_VARIABLE_VALUE_OF"/>
  <SourceLine classname="com.ms.sigada.view.ManterAlunoForm$5" start="311" end="311" startBytecode="29" endBytecode="29" sourcefile="M
  <SourceLine classname="com.ms.sigada.view.ManterAlunoForm$5" start="311" end="311" startBytecode="29" endBytecode="29" sourcefile="M
  <Property name="edu.umd.cs.findbugs.detect.RefComparisonWarningProperty.EMPTY_AND_UNKNOWN" value="true"/>
</BugInstance>
```

Plano de controle de qualidade

- Definição de uma equipe responsável
 - Pontos de verificação/saída
 - Revisão e classificação dos itens analisados
 - Verificação e validação dos requisitos
 - Coleta de dados
 - Suporte/tratamento de erros
- 

Estimativa de custo

Software Size Sizing Method

Unadjusted
Function Points Language

Software Scale Drivers

Precedentedness	<input type="text" value="High"/>	Architecture / Risk Resolution	<input type="text" value="Nominal"/>	Process Maturity	<input type="text" value="Low"/>
Development Flexibility	<input type="text" value="Nominal"/>	Team Cohesion	<input type="text" value="High"/>		

Software Cost Drivers

Product

Required Software Reliability	<input type="text" value="High"/>
Data Base Size	<input type="text" value="Nominal"/>
Product Complexity	<input type="text" value="Low"/>
Developed for Reusability	<input type="text" value="Nominal"/>
Documentation Match to Lifecycle Needs	<input type="text" value="Nominal"/>

Personnel

Analyst Capability	<input type="text" value="Nominal"/>
Programmer Capability	<input type="text" value="Nominal"/>
Personnel Continuity	<input type="text" value="Nominal"/>
Application Experience	<input type="text" value="Low"/>
Platform Experience	<input type="text" value="Low"/>
Language and Toolset Experience	<input type="text" value="Low"/>

Platform

Time Constraint	<input type="text" value="High"/>
Storage Constraint	<input type="text" value="Nominal"/>
Platform Volatility	<input type="text" value="Nominal"/>

Project

Use of Software Tools	<input type="text" value="High"/>
Multisite Development	<input type="text" value="Nominal"/>
Required Development Schedule	<input type="text" value="Nominal"/>

Estimativa de Custo

Cost per Person-Month (Dollars)

Results

Software Development (Elaboration and Construction)

Effort = 42.6 Person-months

Schedule = 12.7 Months

Cost = \$85186





















Total Equivalent Size = 9434 SLOC


Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	2.6	1.6	1.6	\$5111
Elaboration	10.2	4.7	2.2	\$20445
Construction	32.4	7.9	4.1	\$64742
Transition	5.1	1.6	3.2	\$10222

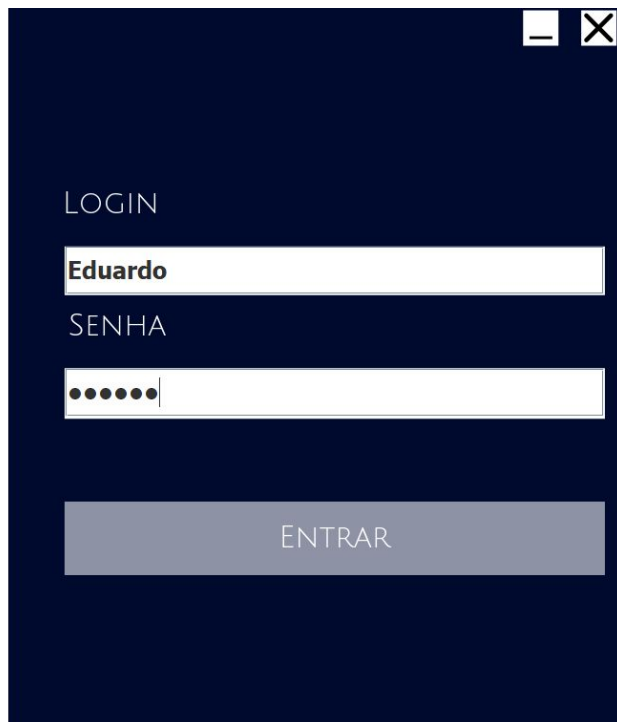
Implementação

MVC na prática

- ▼  > Projeto_InstitutoAdaLovelace [Projeto_InstitutoAdaLovelace m
 - ▶  JRE System Library [JavaSE-1.8]
 - ▼  src
 - ▶  (default package)
 - ▼  com.ms.sigada.bd
 - ▶  package-info.java
 -  Aluno.txt
 -  Contas.txt
 - ▼  com.ms.sigada.controller
 - ▶  AlunoController.java
 - ▶  LoginController.java
 - ▼  com.ms.sigada.model
 - ▶  Aluno.java
 - ▶  AlunoDAO.java
 - ▶  Biblioteca.java
 - ▶  Boletim.java
 - ▶  Coordenador.java
 - ▶  Disciplina.java
 - ▶  Exemplar.java
 - ▶  Extensao.java

- ▶  Livro.java
- ▶  Login.java
- ▶  LoginDAO.java
- ▶  Pessoa.java
- ▶  Professor.java
- ▶  Refeitório.java
- ▶  Sala.java
- ▶  Setor.java
- ▶  Turma.java
- ▼  com.ms.sigada.teste
 - ▶  AlunoDAOTeste.java
 - ▶  ManterAlunoFormTestes.java
- ▼  com.ms.sigada.view
 - ▶  LoginForm.java
 - ▶  ManterAlunoForm.java
 - ▶  MenuRHForm.java
 - ▶  package-info.java

O sigAda

A dark blue login window with a title bar containing minimize and close buttons. The window has the following elements:

- Label: LOGIN
- Username field: A white input field containing the text "Eduardo".
- Label: SENHA
- Password field: A white input field with masked characters (dots).
- Button: A light gray button labeled "ENTRAR".

O sigAda

Bem vindo,

Eduardo



< PARA COMEÇAR, ESCOLHA UMA DAS
CATEGORIAS ABAIXO >



FUNCIONÁRIO



ALUNO

O sigAda

Cadastrar

Consultar

ALUNOS

<

—

×

NOME COMPLETO

CPF

DATA DE NASCIMENTO

NOME RESPONSÁVEL 1

NOME RESPONSÁVEL 2

TELEFONE RESPONSÁVEL 1

TELEFONE RESPONSÁVEL 2

ENDEREÇO 1

ENDEREÇO 2

SALVAR

LIMPAR

O sigAda

Cadastrar

Consultar

ALUNOS

<

—

×

NOME COMPLETO

Monica Santos

CPF

81011035030

DATA DE NASCIMENTO

20/01/2006

NOME RESPONSÁVEL 1

Maria Santos

NOME RESPONSÁVEL 2

Jonas Lima

TELEFONE RESPONSÁVEL 1

2459873857

TELEFONE RESPONSÁVEL 2

2489782388

ENDEREÇO 1

Estrada da Ponte, 120. Centro.

ENDEREÇO 2

SALVAR

LIMPAR

O sigAda

Cadastrar

Consultar

ALUNOS

<

_

X

SELECIONAR ALUNO: Monica Santos - 14

NOME COMPLETO

Monica Santos

CPF

81011035030

DATA DE NASCIMENTO

20/01/2006

NOME RESPONSÁVEL 1

Maria Santos

NOME RESPONSÁVEL 2

Jonas Lima

TELEFONE RESPONSÁVEL 1

2459873857

TELEFONE RESPONSÁVEL 2

2489782388

ENDEREÇO 1

Estrada da Ponte, 120. Centro.

ENDEREÇO 2

-

SALVAR

LIMPAR

DELETAR

Dificuldades e soluções gerais




GAMBIARRA

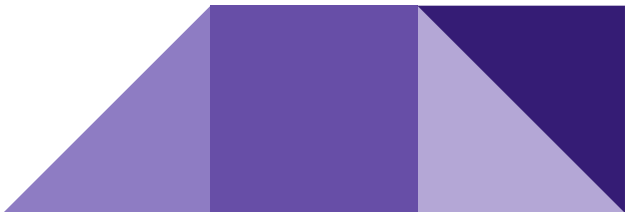
PROGRAMADOR

**PADRÃO
MVC**

Dificuldades

- Primeira vez fazendo um projeto desse tipo
 - Tentativa de aplicação fiel da teoria em uma situação inviável para tal
 - Ausência de um cliente real
 - Ordem dos documentos
 - Uso e aprendizado das ferramentas
- 

Dificuldades

- Dificuldade de definir os tipos de testes (sistema, integração, conformidade e release)
 - Literatura contraditória
 - Ausência de um espaço físico nosso para o desenvolvimento
 - Dificuldade em pôr em prática as definições das funções de equipe feitas no início do projeto
- 

Soluções

- Consultas com profissionais da área
 - Reuso de código/modelos familiares quando possível
 - Gerenciamento de versões
 - Exemplos reais e templates com auxílio dos materiais de apoio
 - Tirar o possível da metodologia, usando elas como um guia
 - Pipefy e Discord
 - Prática
- 