



SISTEMA DE MOEDA ESTUDANTIL

AUGUSTO DE SOUZA - DAVI AGUILAR - PEDRO MOREIRA - RAFAEL CHEQUER

OBJETIVO DO SISTEMA

Estimular o reconhecimento de mérito estudantil por meio de uma moeda virtual, possibilitando trocas por vantagens oferecidas por empresas parceiras.



DESCRIÇÃO GERAL

1.

USUÁRIOS PRINCIPAIS

Alunos, Empresas, Professor

2.

FUNCIONAMENTO Empresas

possuem vantagens para serem selecionadas resgatadas por alunos. Professores distribuem pontos de acordo com indicadores de desempenho arbitrários.

3.

DIFERENCIAIS

Manipulação especificada de vantagens com adição e remoção

ARQUITETURA E TECNOLOGIAS UTILIZADAS



1.

SUB-ARQUITETURA MVC

Separação de camadas de
apresentação, negócio
persistência.



2.

FRAMEWORKS Spring Boot,

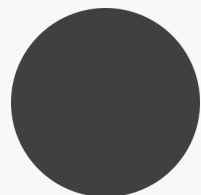
Thymeleaf, H2
Database para armazenamento
temporário.



3.

PERSISTÊNCIA DE DADOS

Banco H2 (in-memory), uso de
JPA/Hibernate.



CLASSES

```
public class Aluno {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id;  
    private String nome;  
    private String email;  
    private String cpf;  
    private String rg;  
    private String endereco;  
    private String instituicaoEnsino;  
    private String curso;  
}
```

ALUNO

```
public class Empresa {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id;  
    private String nome;  
    @OneToMany(mappedBy = "empresa", cascade = CascadeType.ALL)  
    private List<Vantagem> vantagens;  
}
```

EMPRESA

```
public class Vantagem {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id;  
    private String descricao;  
    private double custo;  
    @Lob  
    private String fotoProduto;  
    @ManyToOne  
    @JoinColumn(name = "empresa_id")  
    private Empresa empresa;  
}
```

VANTAGEM

Empresa → Vantagem (One-to-Many)

CLASSES

```
public class Professor {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id;  
    private String nome;  
    private String email;  
    private String cpf;  
    private String departamento;  
    private Integer moedas;  
    private String senha;  
}
```

PROFESSOR

```
public class Extrato {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id;  
    private String data;  
  
    @ManyToOne  
    private Professor professor;  
  
    @ManyToOne  
    private Aluno aluno;  
  
    private String descricao;  
    private Integer valor;  
}
```

EXTRATO

```
class VantagensResgatadas {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id;  
  
    @ManyToOne  
    @JoinColumn(name = "aluno_id")  
    private Aluno aluno;  
  
    @ManyToOne  
    @JoinColumn(name = "vantagem_id")  
    private Vantagem vantagem;  
  
    private double custo;  
  
    @Lob  
    private String fotoVantagem;  
}
```

**VANTAGEM
RESGATADA**

ARQUITETURA MVC

- Organização e Manutenção
- Escalabilidade
- Colaboração
- Segurança e Controle
- Exemplo no Projeto: Controllers gerenciam operações (ex.: CRUD de Aluno), o Model lida com dados, e o View exibe tudo ao usuário.

```
org.example.moedaestudentecombd
├── config
│   └── DataLoader
├── controller
│   ├── AlunoController
│   ├── EmpresaController
│   ├── ProfessorController
│   └── VantagemController
├── model
│   ├── Aluno
│   ├── Empresa
│   ├── Extrato
│   ├── Professor
│   ├── Vantagem
│   └── VantagensResgatadas
├── repository
│   ├── AlunoRepository
│   ├── EmpresaRepository
│   ├── ExtratoRepository
│   ├── ProfessorRepository
│   ├── VantagemRepository
│   └── VantagensResgatadasRepository
├── service
│   ├── AlunoService
│   ├── EmpresaService
│   ├── ExtratoService
│   ├── ProfessorService
│   └── VantagemService
└── App
```

FLUXO DE CADASTRO E OPERAÇÕES CRUD

- Cadastro, listagem, edição e exclusão.
- Thymeleaf para formulários e listagens.
- AlunoRepository,
EmpresaRepository,
VantagemRepository,
ExtratoRepository

**CRUD PARA ALUNO E EMPRESA
PARCEIRA**

**INTEGRAÇÃO COM A
INTERFACE**

REPOSITÓRIOS (JPA)



FUNCIONALIDADES DO SISTEMA

- Cadastro e Login: Alunos, empresas e professores cadastram-se.
- Distribuição de Moedas: Professores distribuem moedas com mensagem de reconhecimento.
- Resgate de Vantagens: Alunos trocam moedas por vantagens.
- Registro de extrato: As transações de alunos e professores são registradas.



GITHUB

1

AUGUSTO

Documentação
Front e Back

2

DAVI

Documentação
Front e Back

3

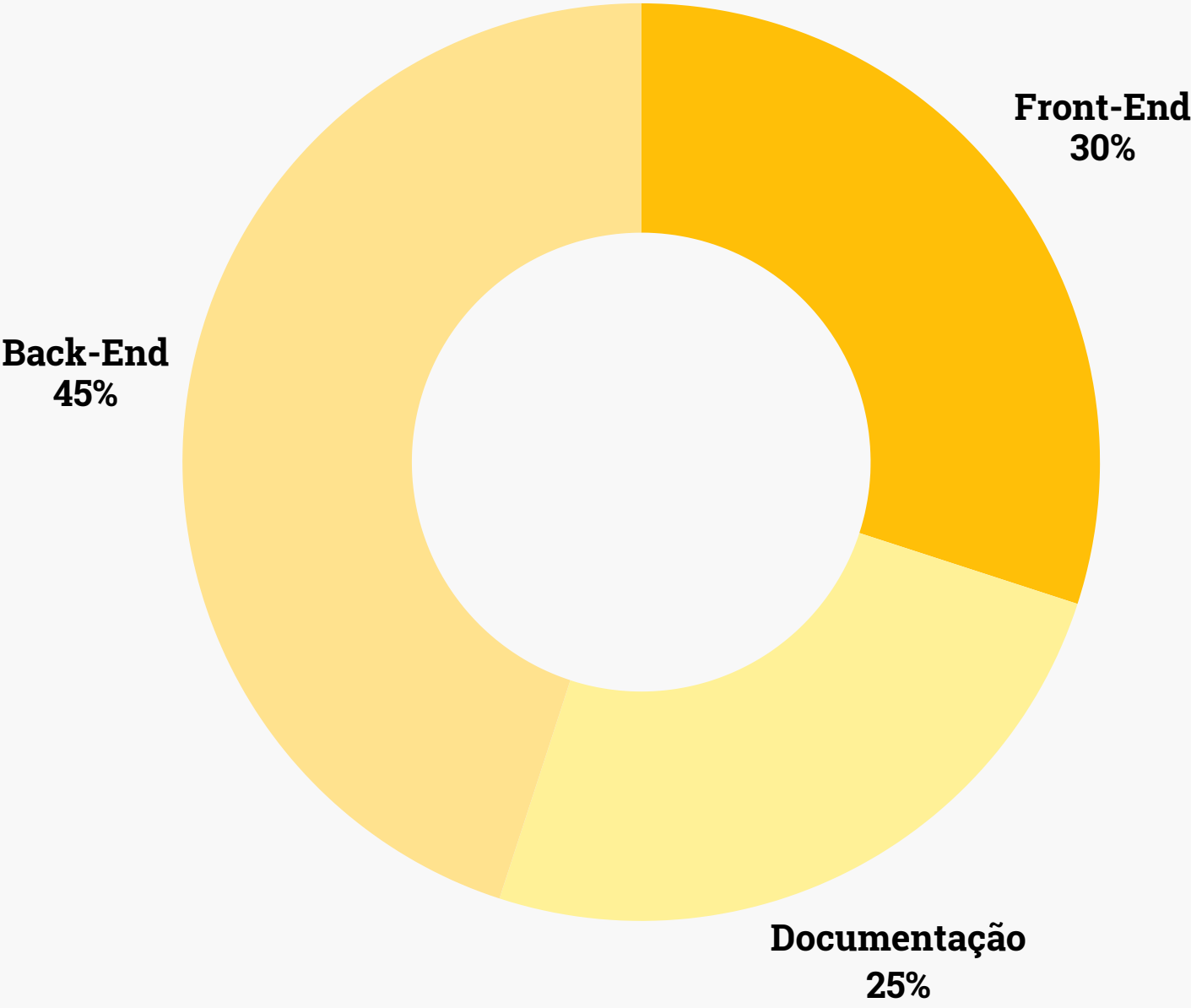
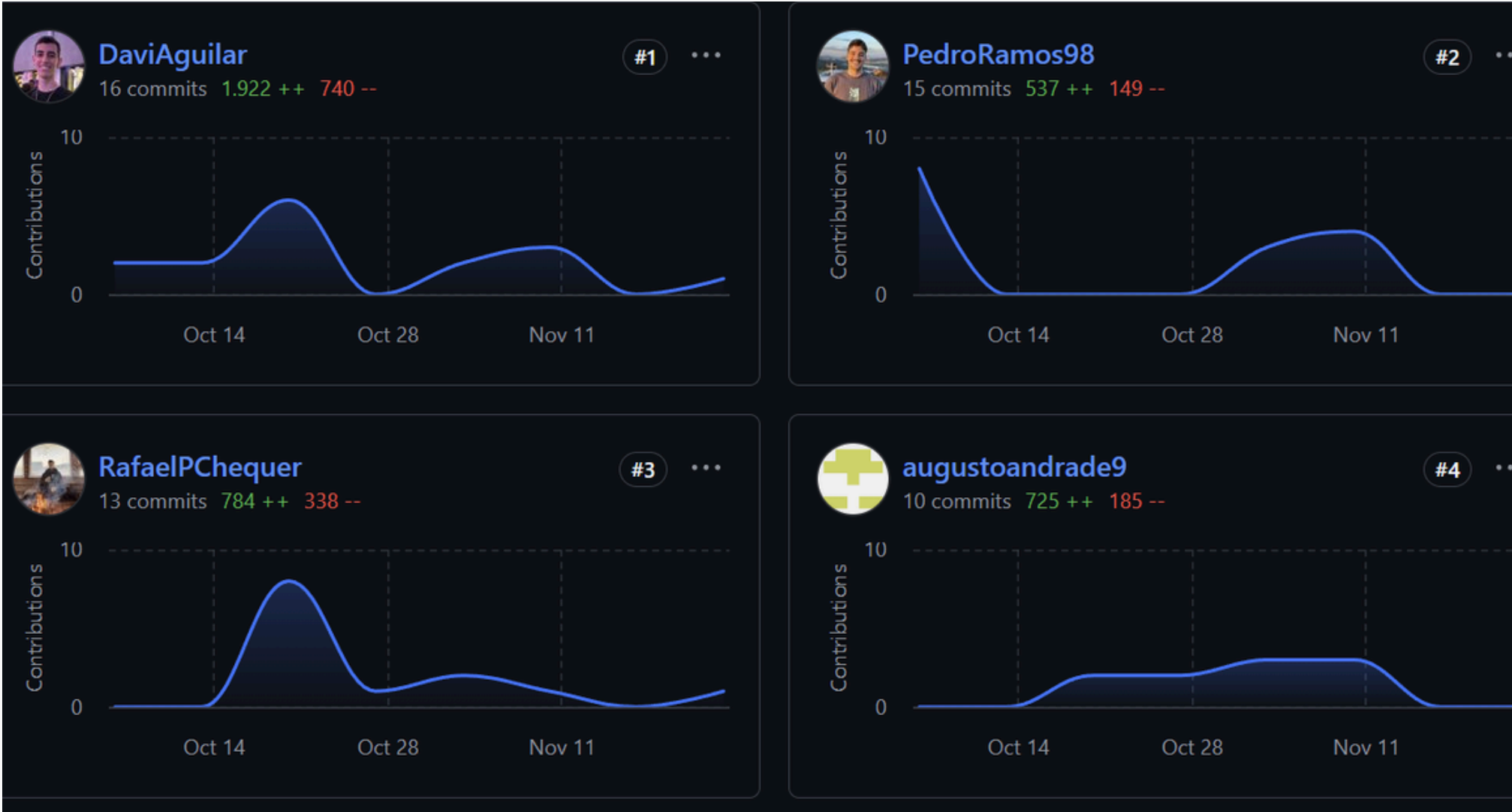
PEDRO

Documentação
Front e Back

4

RAFAEL

Documentação
Front e Back





OBRIGADO