

Gestão de Academia

Módulo - Cadastro

APRESENTAÇÃO DO SOFTWARE

- O objetivo do sistema é efetuar um cadastro de usuários, alunos e funcionários, endereços bem como efetuar a gestão dos serviços oferecidos na forma de planos.
- Através de uma interface simples ele centraliza tarefas de controle de acesso, cadastro e segurança de maneira objetiva, tornando-se uma ferramenta ideal para o controle dos ativos de sua academia.

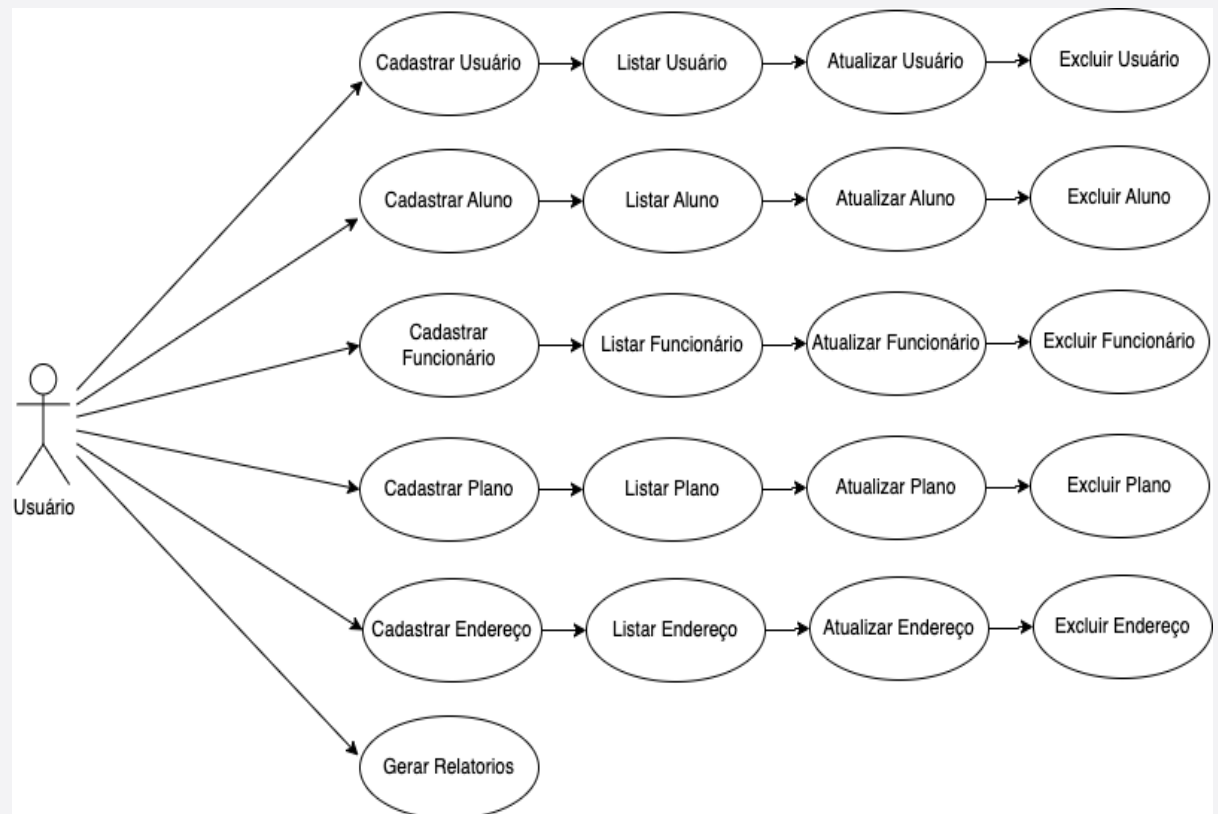


FIGURA 1- MODELO CASO DE USO

Requisitos do Sistema – Boas Práticas

1. Cadastro de Usuário:

- O sistema deve permitir o cadastro de usuários, incluindo informações como usuário, senha, nome, e-mail, tipo de documento, número do documento, data de nascimento.
- Deve ser possível diferenciar entre usuários comuns, alunos e funcionários.



2. Cadastro de Aluno:

- O sistema deve ter a capacidade de cadastrar informações específicas de alunos, como nome, tipo de documento, número do documento, telefone, gênero, data de nascimento.

3. Cadastro de Funcionário:

- Deve ser possível cadastrar informações sobre os funcionários da academia, nome, tipo de documento, número do documento, telefone, e-mail, data nascimento, gênero, data da contratação, data de desligamento, cargo.

4. Cadastro de Plano:

- O sistema deve permitir o cadastro de diferentes planos de academia, especificando características como nível, tipo de plano (mensal, trimestral, anual) e valor.

Requisitos do Sistema – Boas Práticas

5. Cadastro de Endereço:

- Deverá ser possível cadastrar endereços associados aos usuários, alunos, funcionários e à própria academia.
- Campos como rua, número, complemento, CEP, bairro, cidade e estado devem ser contemplados.

6. Relatórios:

- O sistema deve oferecer a capacidade de gerar relatórios diversos, como:
 - Relatório de funcionário vs cargos
 - Relatório de alunos matriculado por ano/mês
 - Relatório de maiores salario por setor
 - Relatório de aniversariantes do mês
 - Outros relatórios específicos conforme as necessidades da academia.



Modelo Conceitual

- O objetivo é a modelagem do banco de dados sua entidade de relacionamento em termos de regras logicas e de negócio.
- Neste fluxo percebemos o esquema do banco onde as entidades se conectam.

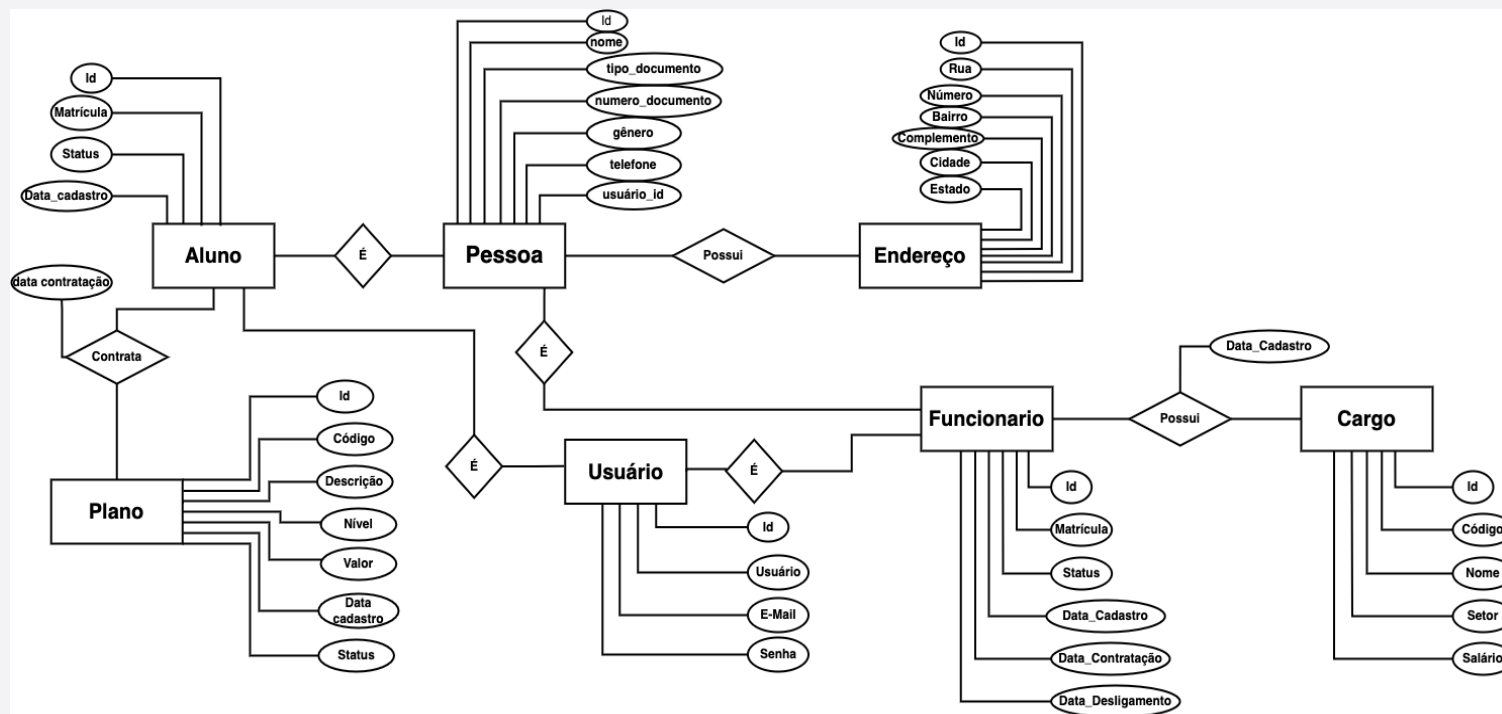


FIGURA 1 - MODELO CONCEITUAL

Modelo Lógico

- O objetivo é a modelagem do banco de dados, sua entidade de relacionamento em termos de regras logicas e de negócio.
- Neste fluxo percebemos o esquema do banco onde as entidades se conectam.

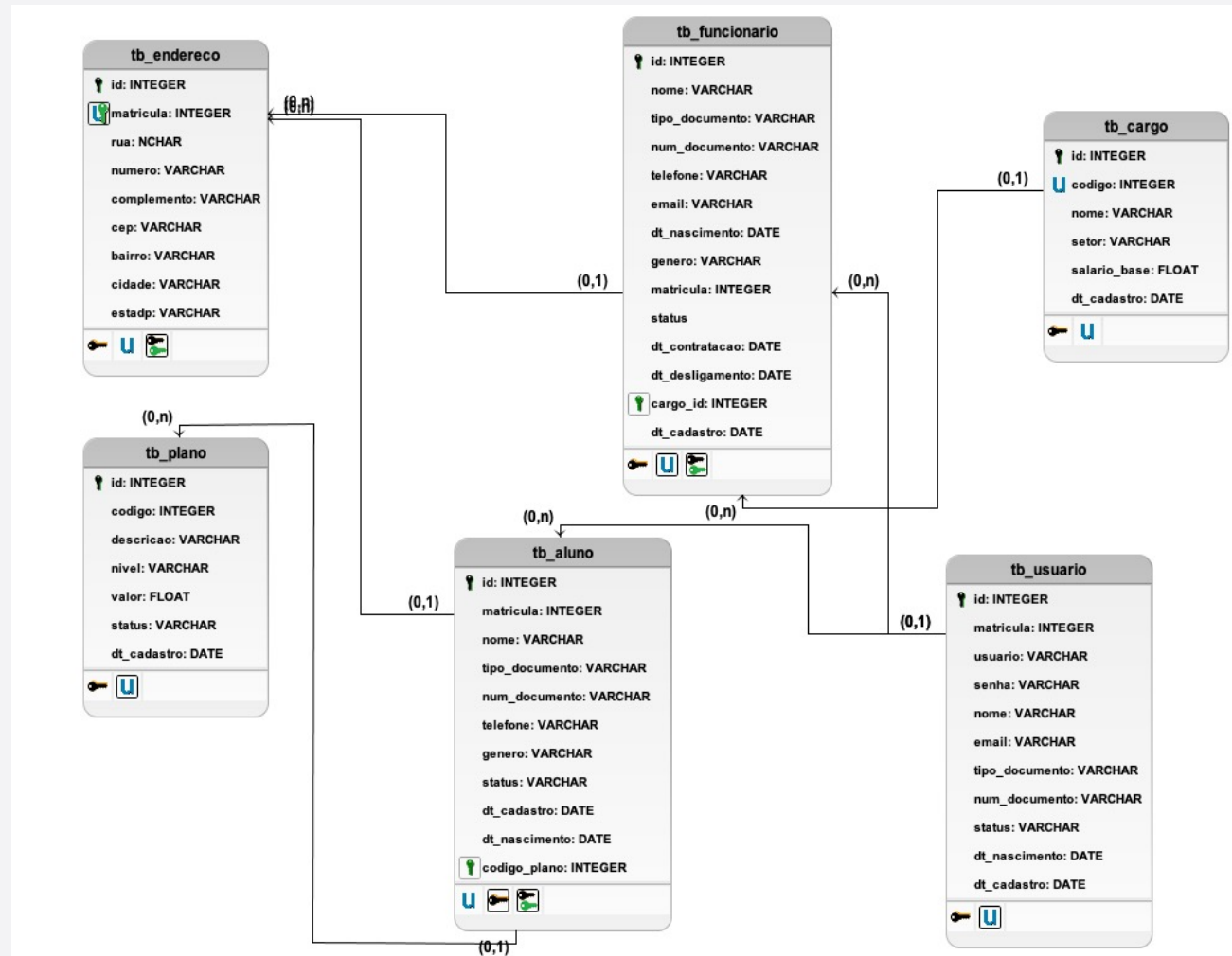


FIGURA 2 - MODELO LOGICO

Diagrama Classes

- O objetivo do diagrama de classe é modelar os objetos que compõe o sistema.
- Nele está contido as Classes, atributos e métodos que serão utilizados no sistema.

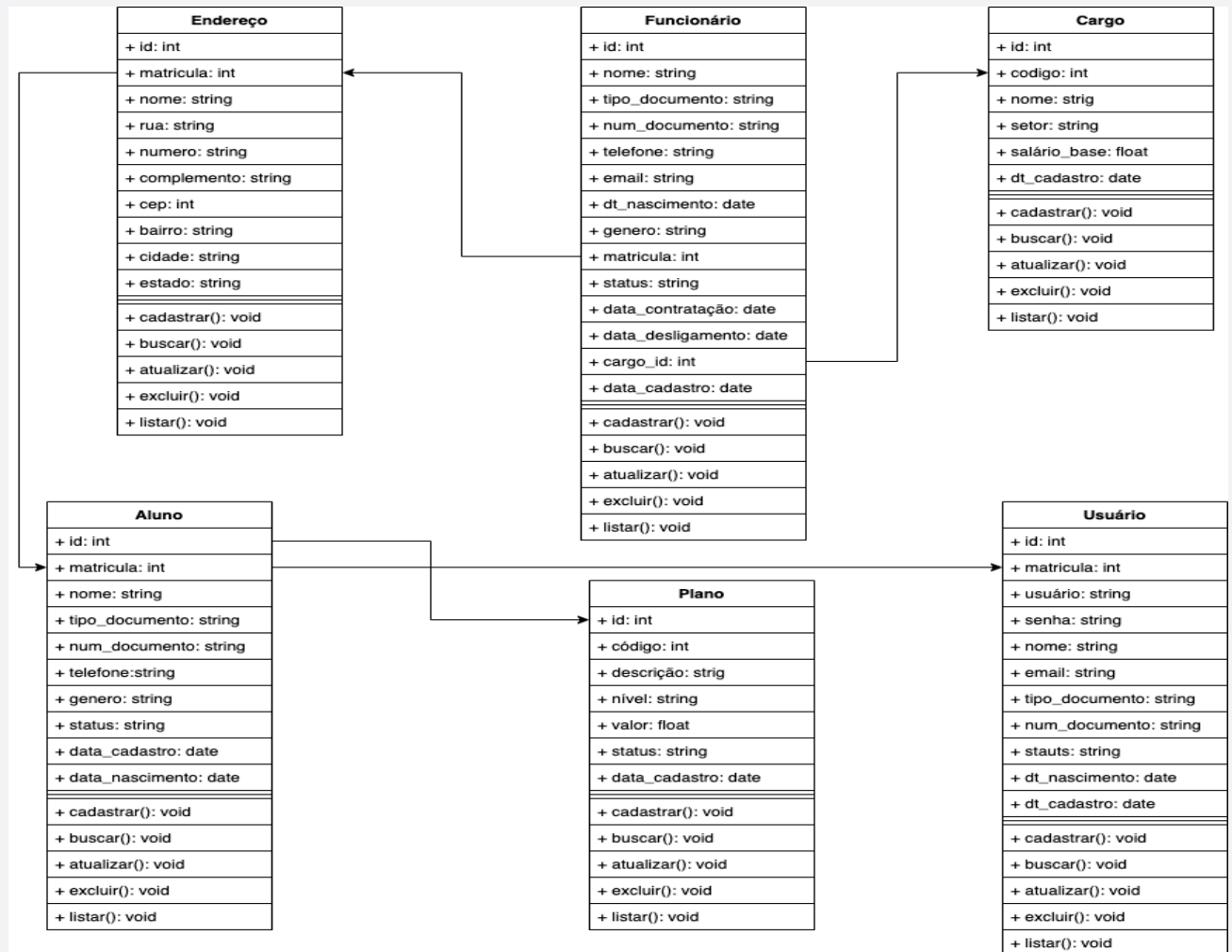


FIGURA 2- MODELO DO DIAGRAMA DE CLASSES

Diagrama do Banco de Dados

- O objetivo é a modelagem do banco de dados sua entidade de relacionamento em termos de regras lógicas e de negócio.
- Neste fluxo percebemos o esquema do banco onde as entidades se conectam.

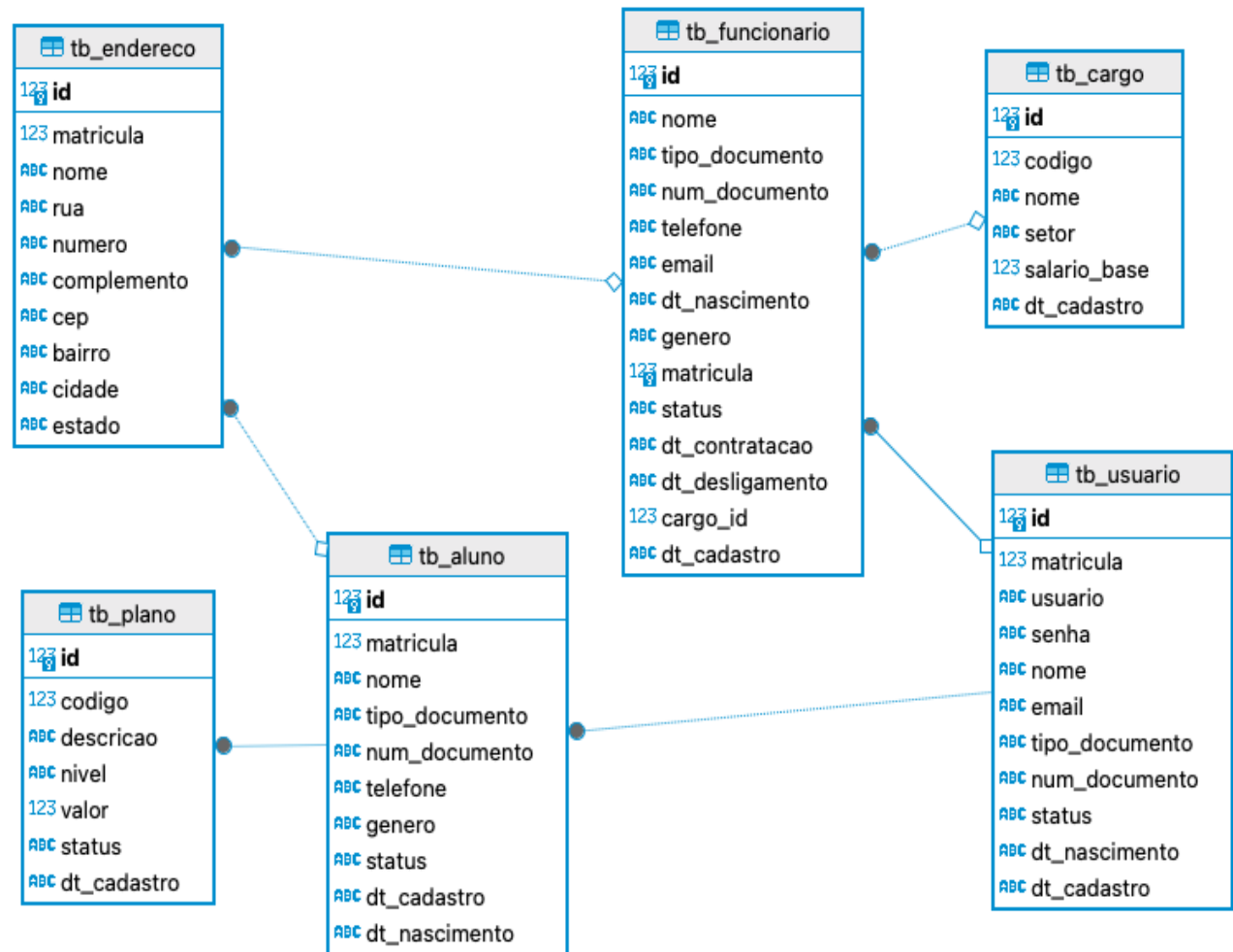
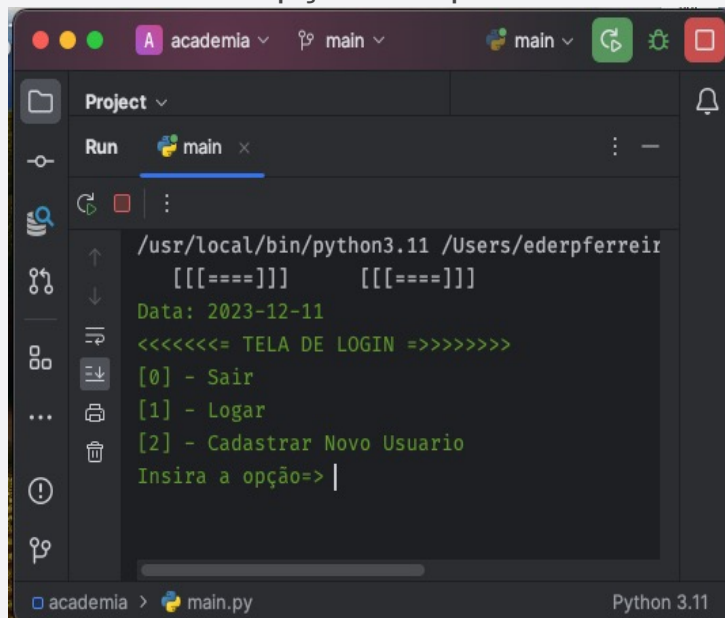


FIGURA 4 - MODELO DE BANCO DE DADOS

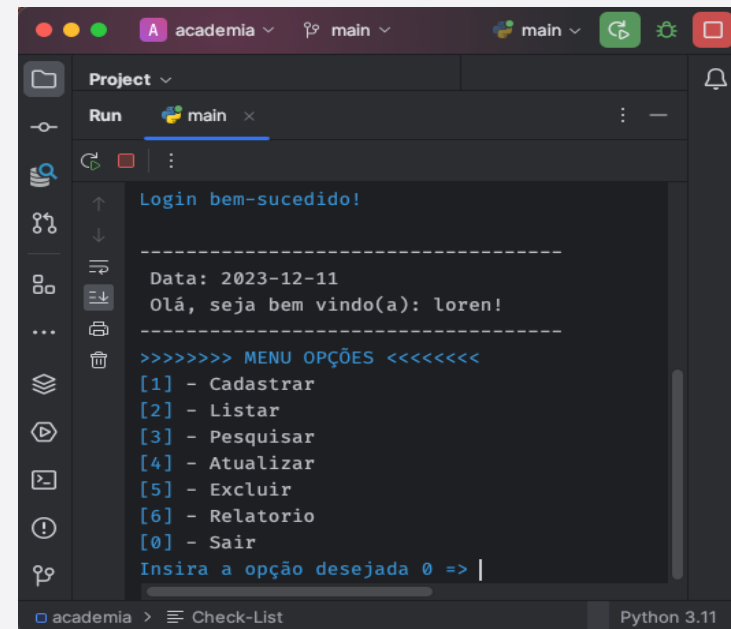
TELAS LOGIN E LOGADO

- Tela de Login: O sistema conta com uma tela de login para autenticação do usuário;
- Tela de logado: Após a validação do login e senha, é liberado o ambiente de logado com o menu de opções disponíveis ao usuário.



```
/usr/local/bin/python3.11 /Users/ederpferreir
[[[===]]] [[[===]]]
Data: 2023-12-11
<<<<<<<= TELA DE LOGIN =>>>>>>>
[0] - Sair
[1] - Logar
[2] - Cadastrar Novo Usuario
Insira a opção=> |
```

FIGURA 5 - TELA DE LOGIN



```
Login bem-sucedido!
-----
Data: 2023-12-11
Olá, seja bem vindo(a): loren!
-----
>>>>>>> MENU OPÇÕES <<<<<<<
[1] - Cadastrar
[2] - Listar
[3] - Pesquisar
[4] - Atualizar
[5] - Excluir
[6] - Relatorio
[0] - Sair
Insira a opção desejada 0 => |
```

FIGURA 6 - TELA DE LOGADO

CADASTRO DE USUÁRIOS

- Cadastro de Usuários: permite cadastrar novos usuários para acesso ao sistema.
- Registra informações básicas para acessar o sistema:

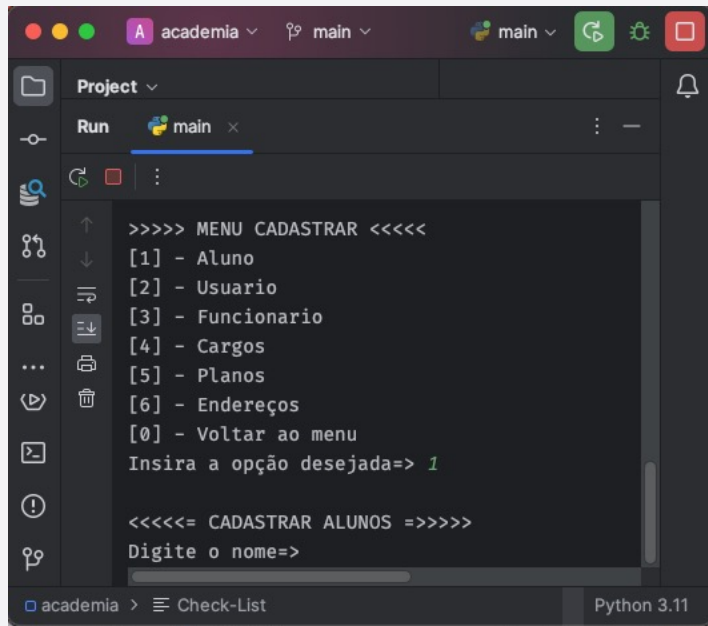
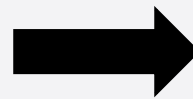


FIGURA 7 - TELA DE CADASTRO DE ALUNOS



tb_usuario	id
	matricula
	usuario
	senha
	nome
	email
	tipo_documento
	num_documento
	status
	dt_nascimento
	dt_cadastro

FIGURA 8 - TABELA DO BANCO

CADASTRO DE ALUNOS

- Cadastro de alunos: permite cadastrar novos alunos para gestão dos serviços oferecidos

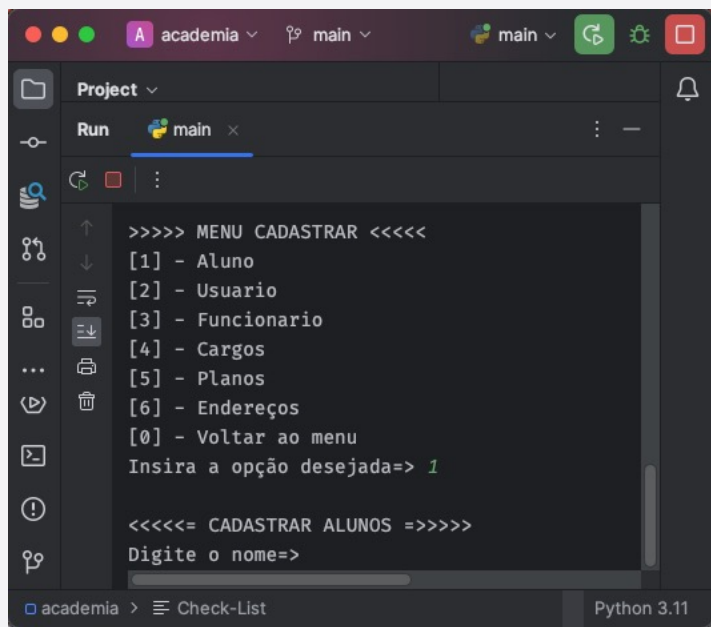


FIGURA 9 - TELA DE CADASTRO DE ALUNOS



tb_aluno	id
	matricula
	nome
	tipo_documento
	num_documento
	telefone
	genero
	status
	dt_cadastro
	dt_nascimento

FIGURA 10 - TABELA DO BANCO

CADASTRO DE FUNCIONÁRIO

- Cadastro de funcionário: permite o cadastro dos funcionários, vinculando cargos, salário e setor.

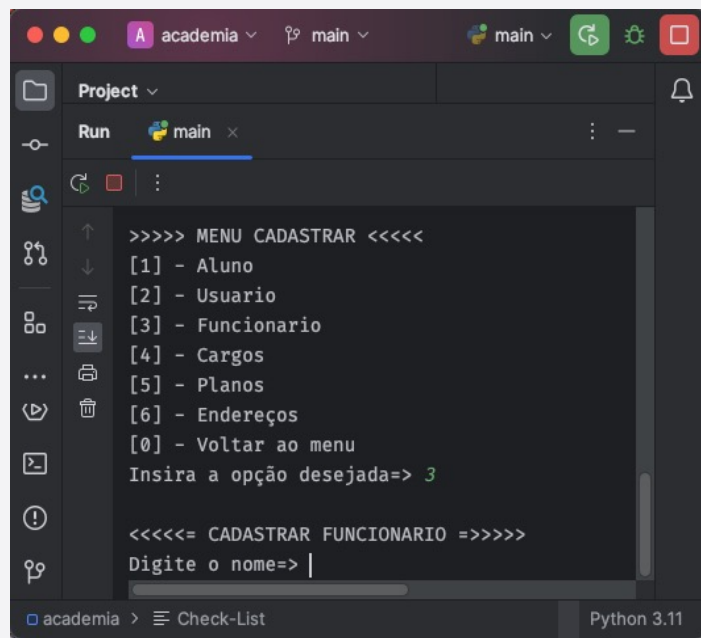
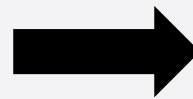


FIGURA 11 - TELA DE CADASTRO DE FUNCIONARIO



tb_funcionario	id
	nome
	tipo_documento
	num_documento
	telefone
	email
	dt_nascimento
	genero
	matricula
	status
	dt_contratacao
	dt_desligamento
	cargo_id
	dt_cadastro

FIGURA 12 - TABELA DO BANCO

CADASTRO DE CARGOS

- Cadastro de cargos e salários : permite o cadastro do cargo, setor e salário base vinculado ao funcionário.

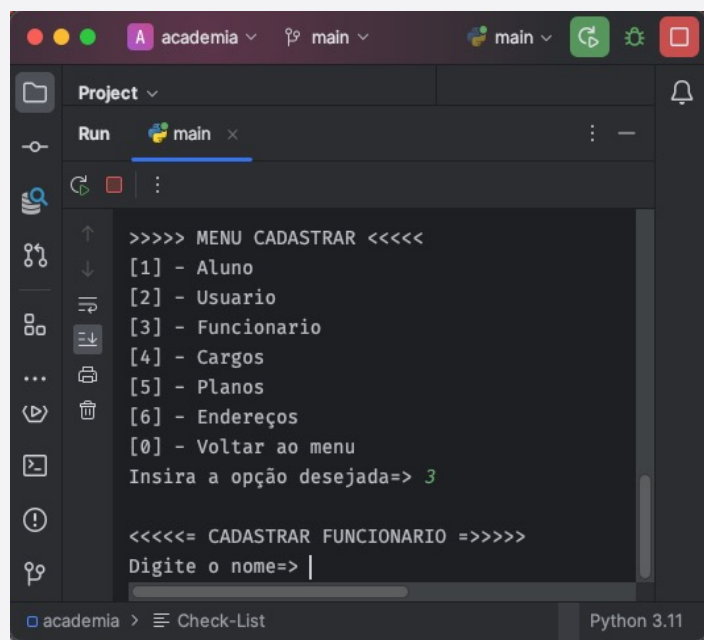


FIGURA 13 - TELA DE CADASTRO DE CARGOS

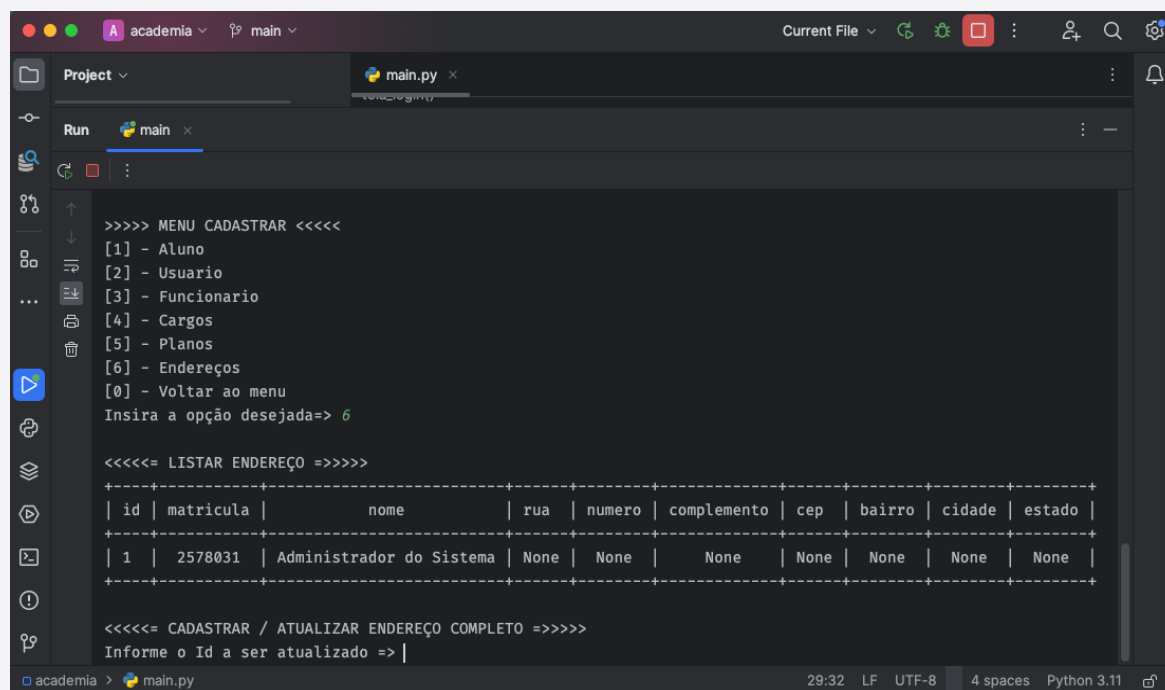


tb_cargo	id
	codigo
	nome
	setor
	salario_base
	dt_cadastro

FIGURA 14 - TABELA DO BANCO

CADASTRO DE ENDEREÇOS

- Cadastro de endereço: permite o cadastro do endereço completo vinculado ao aluno e funcionário.



```
>>>> MENU CADASTRAR <<<<<
[1] - Aluno
[2] - Usuario
[3] - Funcionario
[4] - Cargos
[5] - Planos
[6] - Endereços
[0] - Voltar ao menu
Insira a opção desejada=> 6

<<<<= LISTAR ENDEREÇO =>>>>
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | matricula | nome | rua | numero | complemento | cep | bairro | cidade | estado |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2578031 | Administrador do Sistema | None | None | None | None | None | None | None |
+-----+-----+-----+-----+-----+-----+-----+-----+

<<<<= CADASTRAR / ATUALIZAR ENDEREÇO COMPLETO =>>>>
Informe o Id a ser atualizado => |
```

FIGURA 15 - TELA DE CADASTRO DE ENDEREÇOS

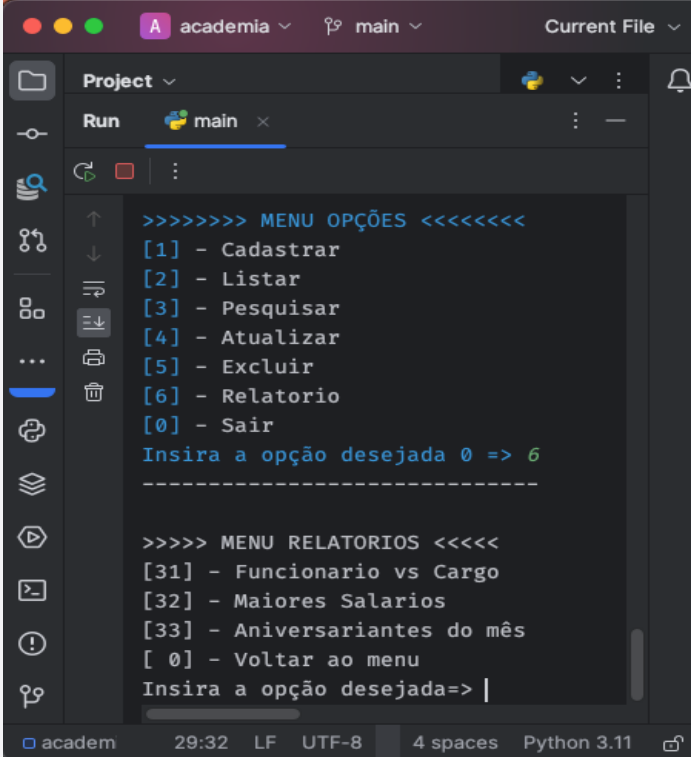


tb_endereco	id
	matricula
	nome
	rua
	numero
	complemento
	cep
	bairro
	cidade
	estado

FIGURA 16 - TABELA DO BANCO

RELATÓRIOS

- Permite a geração dos relatório:
- Funcionário vs Cargos;
- Maiores Salários;
- Aniversariantes do mês



The screenshot shows a terminal window with a dark background. At the top, the window title is 'academia' and the current file is 'main'. The terminal displays two menu options. The first menu, 'MENU OPÇÕES', lists options 1 through 6, with option 6 being 'Relatorio'. The user has entered '6' and the prompt 'Insira a opção desejada 0 => 6' is shown. The second menu, 'MENU RELATORIOS', lists options 31 through 33, with option 33 being 'Aniversariantes do mês'. The prompt 'Insira a opção desejada=> |' is shown at the bottom of the second menu. The terminal status bar at the bottom indicates the file is 'academ', the cursor is at line 29, column 32, the encoding is 'UTF-8', the indentation is '4 spaces', and the Python version is 'Python 3.11'.

```
>>>>>>> MENU OPÇÕES <<<<<<<<
[1] - Cadastrar
[2] - Listar
[3] - Pesquisar
[4] - Atualizar
[5] - Excluir
[6] - Relatorio
[0] - Sair
Insira a opção desejada 0 => 6
-----

>>>>> MENU RELATORIOS <<<<<
[31] - Funcionario vs Cargo
[32] - Maiores Salarios
[33] - Aniversariantes do mês
[ 0] - Voltar ao menu
Insira a opção desejada=> |
```

FIGURA 17 - TELA DE RELATORIOS

RELATÓRIO – Funcionário vs Cargos e Salários

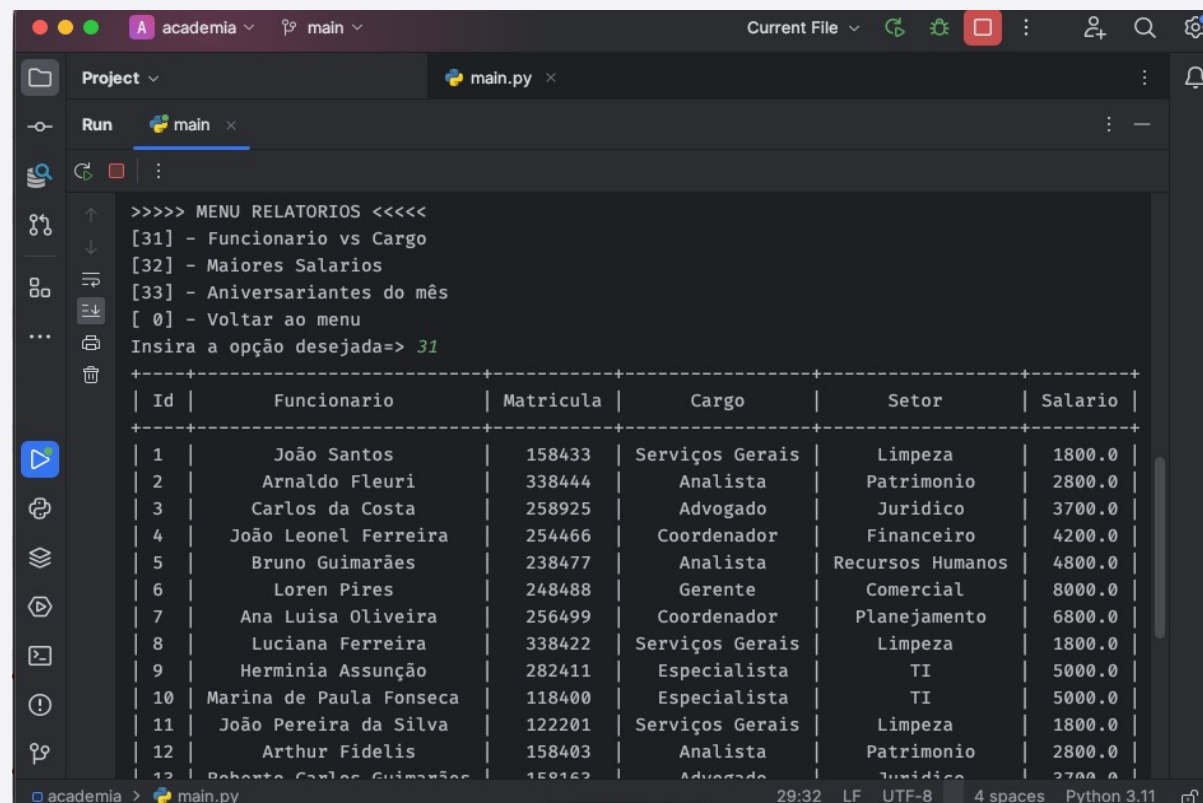
- Funcionários vs cargos: Permite a visualização dos funcionários cadastrados, sua matricula, cargo, setor e salário base cadastrados.
- Método utilizado

SELECT

tb_funcionario.id, tb_funcionario.nome, tb_funcionario.matricula, tb_cargo.nome, tb_cargo.setor, tb_cargo.salario_base

FROM tb_funcionário

INNER JOIN tb_cargo ON
tb_funcionario.cargo_id =
tb_cargo.codigo



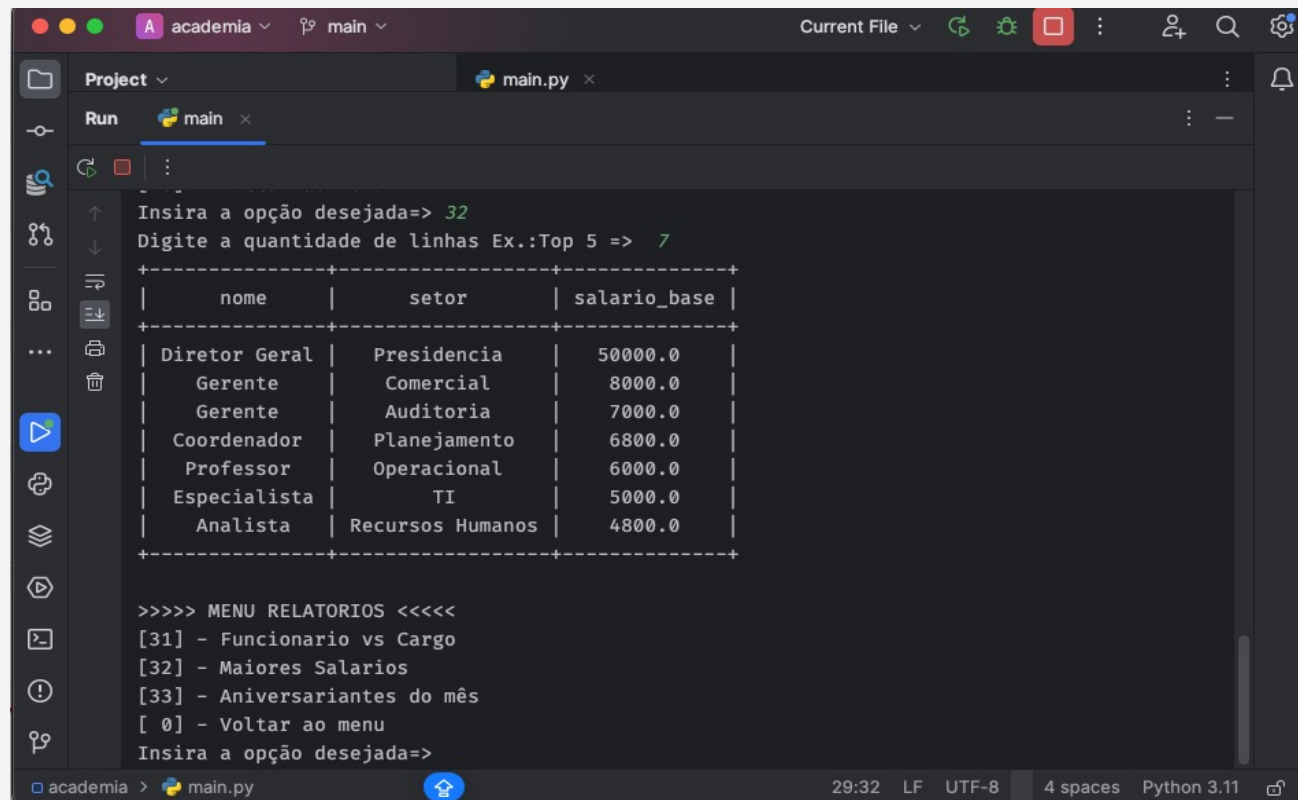
The screenshot shows a Python IDE window titled 'academia' with a file named 'main.py'. The code in the editor displays a menu and a table of employee data. The menu options are: [31] - Funcionario vs Cargo, [32] - Maiores Salarios, [33] - Aniversariantes do mês, and [0] - Voltar ao menu. The user has selected option 31. Below the menu is a table with 6 columns: Id, Funcionario, Matricula, Cargo, Setor, and Salario. The table contains 13 rows of data.

Id	Funcionario	Matricula	Cargo	Setor	Salario
1	João Santos	158433	Serviços Gerais	Limpeza	1800.0
2	Arnaldo Fleuri	338444	Analista	Patrimonio	2800.0
3	Carlos da Costa	258925	Advogado	Juridico	3700.0
4	João Leonel Ferreira	254466	Coordenador	Financeiro	4200.0
5	Bruno Guimarães	238477	Analista	Recursos Humanos	4800.0
6	Loren Pires	248488	Gerente	Comercial	8000.0
7	Ana Luisa Oliveira	256499	Coordenador	Planejamento	6800.0
8	Luciana Ferreira	338422	Serviços Gerais	Limpeza	1800.0
9	Herminia Assunção	282411	Especialista	TI	5000.0
10	Marina de Paula Fonseca	118400	Especialista	TI	5000.0
11	João Pereira da Silva	122201	Serviços Gerais	Limpeza	1800.0
12	Arthur Fidelis	158403	Analista	Patrimonio	2800.0
13	Roberto Carlos Guimarães	158163	Advogado	Juridico	3700.0

FIGURA 18 – TELA DE RELATORIO FUNCIONARIO VS CARGOS E SALARIOS

RELATÓRIO – Top Salários por Cargo e Setor

- Maiores Salários: Permite a visualização dos Top salários por cargo e setor, definindo a quantidade de linhas a ser plotada.



The screenshot shows a Python IDE window with a terminal running a script. The terminal displays a menu with options [31] - Funcionario vs Cargo, [32] - Maiores Salarios, [33] - Aniversariantes do mês, and [0] - Voltar ao menu. Option [32] has been selected, and the user has entered 7 for the number of lines. The terminal then displays a table of salaries for the top 7 positions.

nome	setor	salario_base
Diretor Geral	Presidencia	50000.0
Gerente	Comercial	8000.0
Gerente	Auditoria	7000.0
Coordenador	Planejamento	6800.0
Professor	Operacional	6000.0
Especialista	TI	5000.0
Analista	Recursos Humanos	4800.0

Below the table, the terminal shows the menu again, and the user has entered 32 to select the 'Maiores Salarios' option.

FIGURA 19 – TELA DE RELATORIO DOS MAIORES SALARIOS POR CARGO E SETOR

RELATÓRIO – Gráfico Top Salários por Setor

- Gráfico Maiores Salários por setor: Permite a visualização dos Top salários por cargo e setor, definindo a quantidade de linhas a ser plotada.

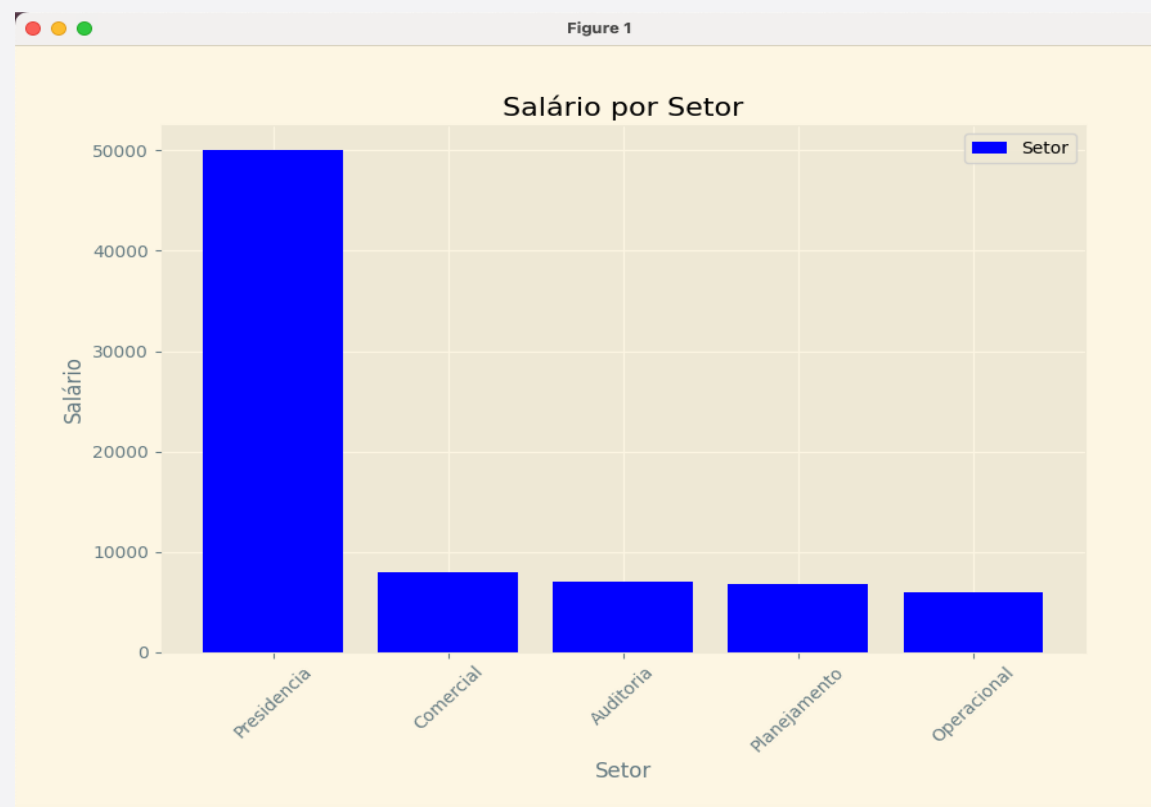
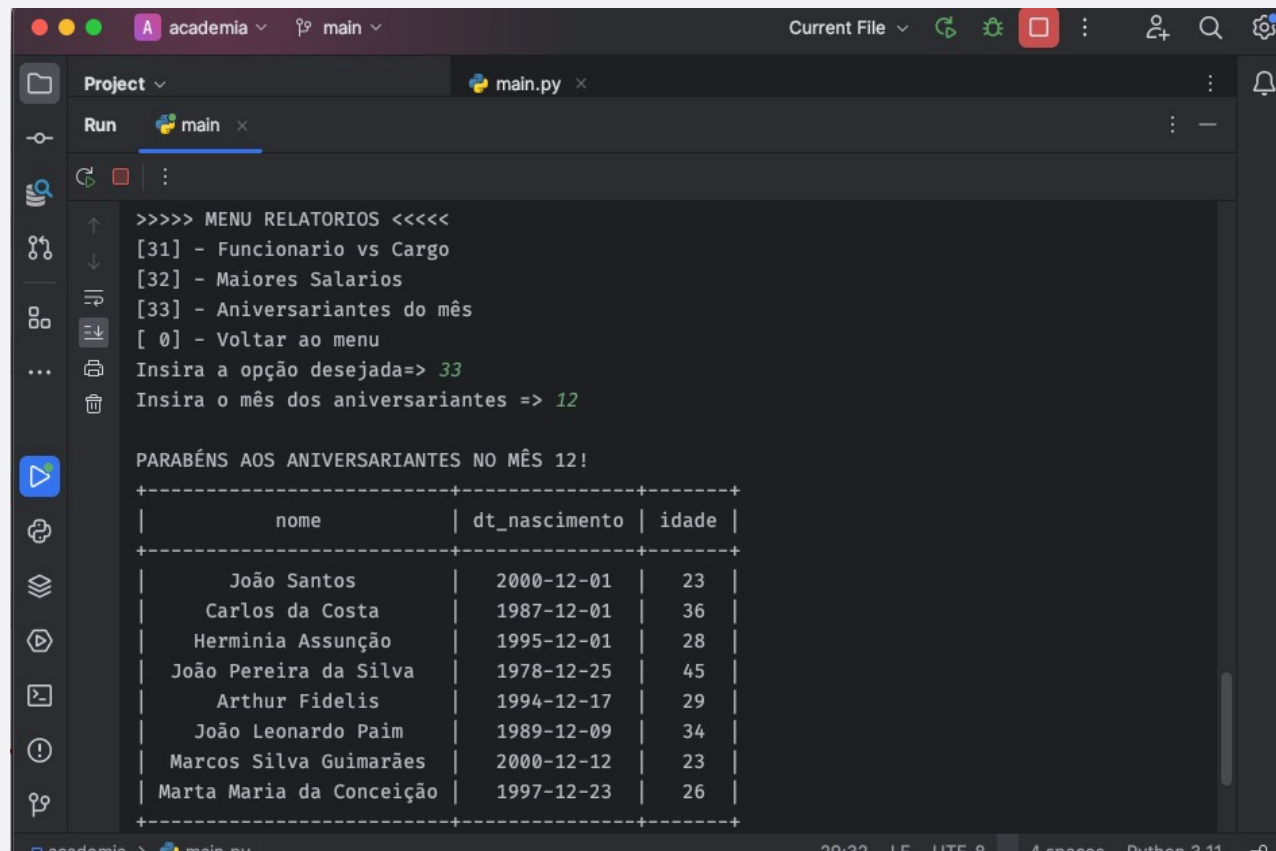


FIGURA 20 – TELA DE RELATORIO DOS MAIORES SALARIOS POR CARGO E SETOR

RELATÓRIO – Aniversariantes do mês

- Aniversariantes do mês: Permite a visualização da relação de aniversariantes a partir da definição do mês.



```
>>>>> MENU RELATORIOS <<<<<
[31] - Funcionario vs Cargo
[32] - Maiores Salarios
[33] - Aniversariantes do mês
[ 0] - Voltar ao menu
Insira a opção desejada=> 33
Insira o mês dos aniversariantes => 12

PARABÉNS AOS ANIVERSARIANTES NO MÊS 12!
+-----+
| nome | dt_nascimento | idade |
+-----+
| João Santos | 2000-12-01 | 23 |
| Carlos da Costa | 1987-12-01 | 36 |
| Herminia Assunção | 1995-12-01 | 28 |
| João Pereira da Silva | 1978-12-25 | 45 |
| Arthur Fidelis | 1994-12-17 | 29 |
| João Leonardo Paim | 1989-12-09 | 34 |
| Marcos Silva Guimarães | 2000-12-12 | 23 |
| Marta Maria da Conceição | 1997-12-23 | 26 |
+-----+
```

FIGURA 21 – TELA DE RELATORIO DOS ANIVERSARIANTES DO MES

RELATÓRIO – Gráfico Alunos Matriculados

- Gráfico Alunos Matriculados: Permite a visualização da quantidade de alunos matriculados por mês, definindo o ano a ser plotado.

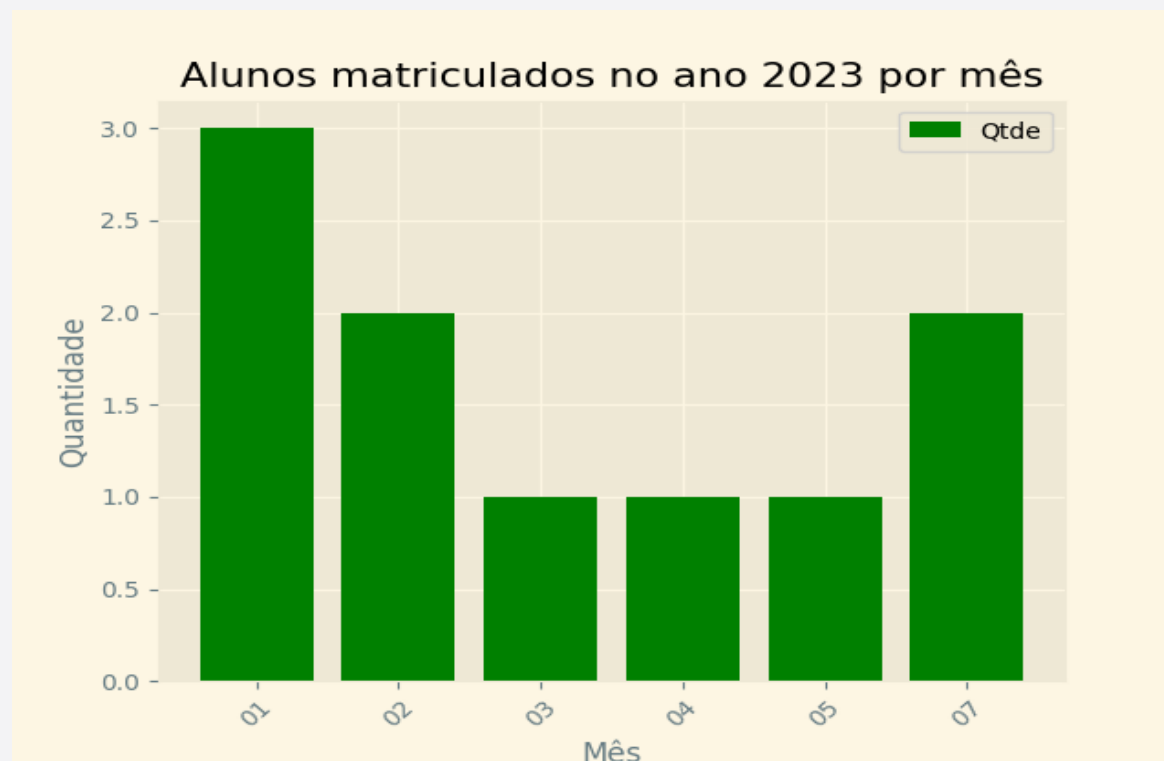


FIGURA 22 – TELA DE RELATORIO ALUNOS MATRICULADOS MES

SQL – Código de criação do Banco e tabelas

```
cria_banco.py x
1  # ----- CRIA BANCO -----
2  import sqlite3
3  con = sqlite3.connect(database: 'Defs/academia.db', check_same_thread=False)
4  cur = con.cursor()
5  print('Banco de dados academia.db criado!!!')
```

FIGURA 23 – SQL DO BANCO DE DADOS ACADEMIA

```
6
7  # ----- CRIA TABELAS -----
8
9  with con:
10     cur.execute('''CREATE TABLE IF NOT EXISTS 'tb_aluno'(
11         'id' INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
12         'matricula' INTEGER NOT NULL UNIQUE,
13         'nome' TEXT,
14         'tipo_documento' TEXT,
15         'num_documento' VARCHAR(10) NOT NULL UNIQUE,
16         'telefone' VARCHAR(12),
17         'genero' VARCHAR(1),
18         'status' TEXT,
19         'dt_cadastro' DATE,
20         'dt_nascimento' DATE,
21         FOREIGN KEY("matricula") REFERENCES "tb_usuario"("id")
22     ''')
```

FIGURA 24 – SQL TB_ALUNO

SQL – Código de criação do Banco e tabelas

```
22 ,
23
24 cur.execute('''CREATE TABLE IF NOT EXISTS 'tb_usuario'(
25     'id' INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
26     'matricula' INTEGER NOT NULL UNIQUE,
27     'usuario' VARCHAR(15),
28     'senha' VARCHAR(10),
29     'nome' VARCHAR(15),
30     'email' VARCHAR(15),
31     'tipo_documento' TEXT,
32     'num_documento' VARCHAR(10) UNIQUE,
33     'status' TEXT,
34     'dt_nascimento' DATE,
35     'dt_cadastro' DATE)
36 ''')
37
```

FIGURA 25 – SQL TB_USUARIO

```
37
38 cur.execute('''CREATE TABLE IF NOT EXISTS 'tb_funcionario'(
39     'id' INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
40     'nome' TEXT,
41     'tipo_documento' TEXT,
42     'num_documento' VARCHAR(10) UNIQUE,
43     'telefone' VARCHAR(12),
44     'email' VARCHAR(15),
45     'dt_nascimento' DATE,
46     'genero' VARCHAR(1),
47     'matricula' INTEGER,
48     'status' TEXT,
49     'dt_contratacao' DATE,
50     'dt_desligamento' DATE,
51     'cargo_id' INTEGER,
52     'dt_cadastro' DATE,
53     FOREIGN KEY("cargo_id") REFERENCES "tb_cargo"("codigo"),
54     FOREIGN KEY("matricula") REFERENCES "tb_usuario"("matricula")
55     )''')
```

FIGURA 26 – SQL TB_FUNCIONARIO

SQL – Código de criação do Banco e tabelas

```
57 cur.execute('''CREATE TABLE IF NOT EXISTS 'tb_cargo'(  
58 'id' INTEGER NOT NULL PRIMARY KEY,  
59 'codigo' INTEGER NOT NULL UNIQUE,  
60 'nome' TEXT,  
61 'setor' TEXT,  
62 'salario_base' FLOAT,  
63 'dt_cadastro' DATE)  
64 ''')  
65  
66 cur.execute('''CREATE TABLE IF NOT EXISTS 'tb_plano'(  
67 'id' INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
68 'codigo' INTEGER NOT NULL UNIQUE,  
69 'descricao' TEXT NOT NULL,  
70 'nivel' TEXT NOT NULL,  
71 'valor' FLOAT,  
72 'status' TEXT,  
73 'dt_cadastro' DATE,  
74 FOREIGN KEY("codigo") REFERENCES "tb_aluno"("matricula"))  
75 ''')
```

FIGURA 27 – SQL TB_CARGO. E TB_PLANO

SQL – Código de criação do Banco e tabelas

```
76
77 cur.execute('''CREATE TABLE IF NOT EXISTS 'tb_endereco'(
78     'id' INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
79     'matricula' INTEGER,
80     'nome' TEXT,
81     'rua' TEXT,
82     'numero' VARCHAR(10),
83     'complemento' TEXT,
84     'cep' VARCHAR(8),
85     'bairro' TEXT,
86     'cidade' TEXT,
87     'estado' TEXT,
88     FOREIGN KEY("matricula") REFERENCES "tb_aluno"("matricula"),
89     FOREIGN KEY("matricula") REFERENCES "tb_funcionario"("matricula")
90 )''')
91
92 con.commit()
93 print('Tabelas Aluno - Usuario - Funcionario - Cargo - Plano - Endereço criadas!!!')
94
95 # ----- FECHA BANCO E CONEXÃO -----
96 cur.close() # FECHA O OBJETO
97 con.close() # FECHA A CONEXÃO
```

FIGURA 28 - SQL TB_ENDERECO

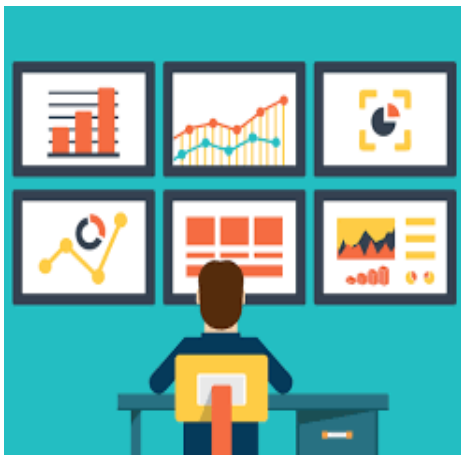
ESPECIFICAÇÕES TÉCNICAS

- Banco de Dados: Padrão do sistema em SQLite, podendo no entanto, ser utilizado com a maioria dos bancos de dados atualmente disponíveis (Oracle, Sybase, Mysql, Firebird, PostgreSQL);
- Linguagem de programação: Python
- IDE: PyCharm CE

MELHORIAS PARA O PROXIMO MÓDULO

1. Criação do módulo Financeiro;
2. Criação do módulo Matricular aluno;
3. Verificação no banco se o registro já existe;
4. Criar função esconder senha digitada;
5. Validação de e-mail;
6. Ajustar o padrão de datas para BR de datas de AAAA-MM-DD para DD-MM-AAAA;
7. Criar opção de 0 pra sair quando iniciado o cadastro;
8. Criar tabela de log no sistema;
9. Esconder a senha digitada;
10. Exportar relatório para PDF;

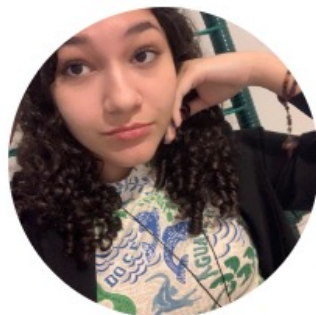




Equipe



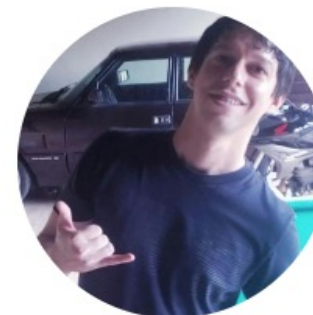
Eder Ferreira



Loren



Guilherme Figueiredo



Marlon Sampaio