

AutoBots

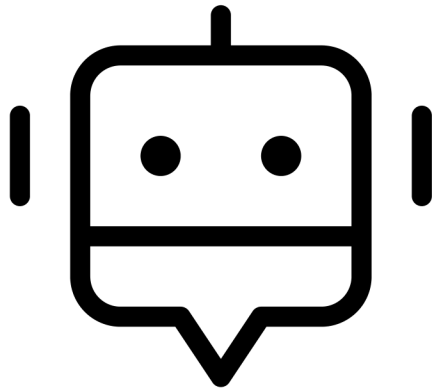
Clube robótica 2023



José R.

Turma: 1BM

Programador do projeto

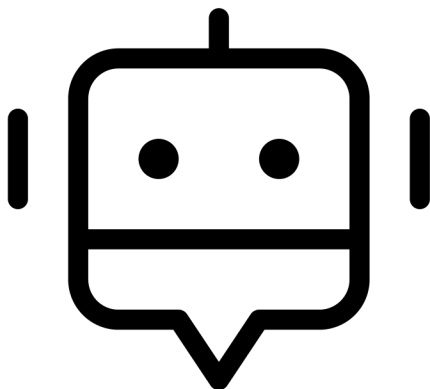




João L.

Turma: 1AD

Apresentador do projeto

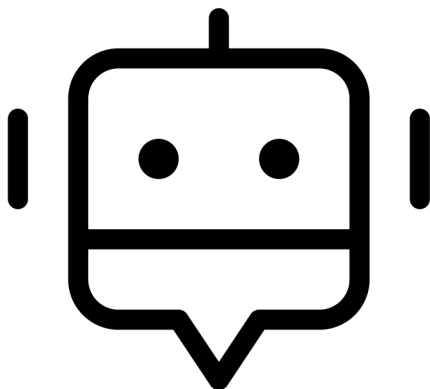


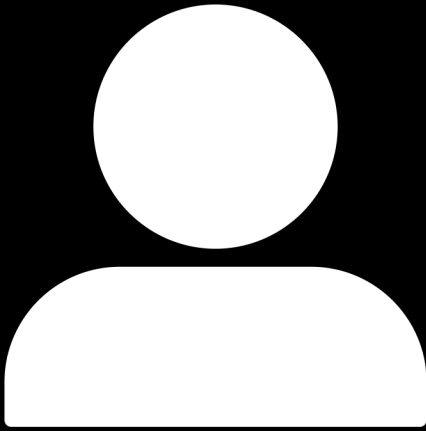


Lílyan M.

Turma: 3AD

Apresentadora do projeto

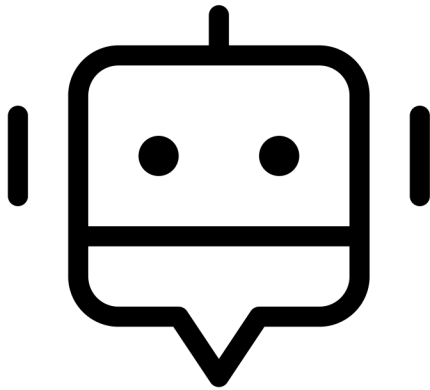


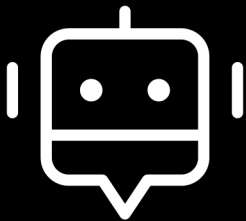


Kaio

Turma:

Apresentador do projeto





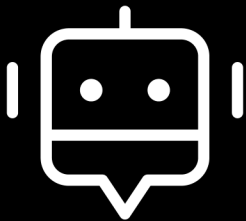
Conheça as tecnologias

SQLite3

Mesmo este projeto não sendo algo profissional, ele foi desenvolvido para revelar a todos como a programação é incrível e fácil. Então para isto o simulador irá usar uma forma de armazenamento de dados chamado SQL (Structured Query Language). Em português, a tradução corresponde a “Linguagem de Consulta Estruturada”.

Porém para ter um SQL, foi usado o SQLite3, um banco de dados que fica armazenado na máquina e não em uma nuvem.

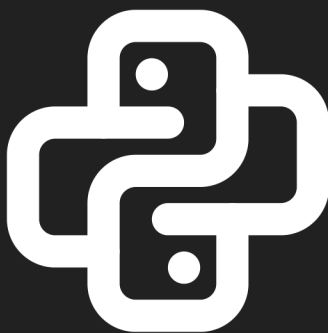


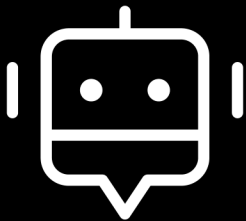


Conheça as tecnologias

Python

Para este projeto, utilizamos a linguagem de programação python, cujo umas das maiores e mais famosas linguagens do mundo. Conhecida pela sua simplicidade e também por conseguir atuar em várias áreas de desenvolvimento, como por exemplo desenvolvimento web, análise de dados e também possível criar robô com python.





Lendo o código

Python

```
> def menu(): # Atual menu do simulador...

> def escolha_menu(): # Responsável pela escolha do usuário...

    # Neste projeto quero que os ids (identificados) tenham 5 dígitos.
    # E esses dígitos sejam todos aleatórios.
> def gerar_id(): # Função para gerar o id (identificador) dos cursos, professores...

> def registrar_aluno(nome, curso): ...
|
> def registrar_professor(nome, idade, salario): ...

> def registrar_curso(nome, instrutor): ...

> def verificar_alunos(): ...

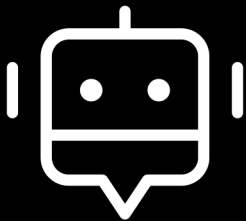
> def verificar_professores(): ...

> def verificar_cursos(): ...

> def procurar_curso(curso_id): ...

while True:
    escolha_menu()
```

O nosso projeto foi desenvolvido com a ideia de ter um código limpo, então para isto ele foi projetado através de ***defs***



Lendo o código

Python

Os **defs** são uma das funções que o próprio possui, é com ele que os programadores criam atalhos para não precisar reescrever o código mais de uma vez. Para serem aplicadas no código, eles precisam ser *puxadas*.

Criando a def do menu

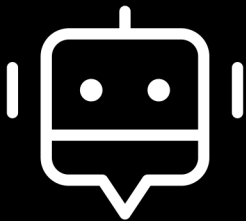
```
def menu(): # Atual menu do simulador
    print('')
    print(' Simulador escolar - 2023')
    print(' 1. Verificar alunos')
    print(' 2. Verificar professores')
    print(' 3. Verificar cursos')
    print(' 4. Registrar aluno')
    print(' 5. Registrar professor')
    print(' 6. Registrar curso')
    print(' 7. Analisar os dados ★')
```

Este é o atalho que irá ficar o nosso menu.

Puxando o menu criado

```
def escolha_menu(): # Responsável pela escolha
    menu()
    escolha = input('Digite o número: ')
```

Agora sim, toda vez que o usuário tiver a opção de escolher algo, o menu irá aparecer.



Lendo o código

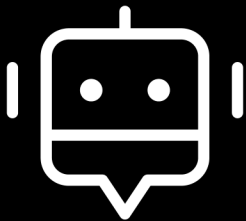
Python

```
def escolha_menu(): # Responsável pela escolha do usuário
    menu()
    escolha = input('Digite o número: ')

    if escolha == '1':
        print('')
        print(verificar_alunos())
        input('PRESSIONE "ENTER" PARA CONTINUAR.')
    elif escolha == '2':
        print('')
        print(verificar_professores())
        input('PRESSIONE "ENTER" PARA CONTINUAR.')
    elif escolha == '3':
        print('')
        print(verificar_cursos())
        input('PRESSIONE "ENTER" PARA CONTINUAR.')
```

Ainda no menu de opção do usuário, iremos fazer a verificação de sua escolha. Os **ifs** são métodos de verificação que o python utiliza.

Traduzindo o código estamos criando uma condição se a escolha for igual ao número 1, irá retornar a lista dos alunos

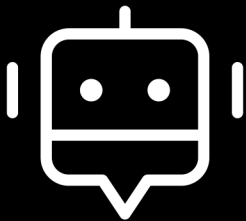


Lendo o código

Python

```
def verificar_alunos():  
    try: # Isto é um tratamento de erro, caso o código dentro d  
        conex = sqlite3.connect('database.db')  
        cursor = conex.cursor()  
  
        cursor.execute('SELECT * FROM alunos')  
  
        alunos_encontrados = '' # String que irá receber o ID e  
  
        for aluno_id, aluno_nome, aluno_curso in cursor.fetchall():  
            alunos_encontrados += f'ID: {aluno_id} | ALUNO: {aluno_nome} | CURSO: {aluno_curso} |  
        return alunos_encontrados  
  
    except Exception as error: # Caso o código dentro do "try"  
        print(f'Erro encontrado: {error}')  
    finally: # Ação que irá acontecer no final, é opcional, afi  
        conex.close()
```

É graças ao **def** que iremos economizar o nosso tempo na hora de verificar os alunos, afinal não iremos rescrever o código toda vez que queremos ver os alunos, é só puxar a função def dele.



Lendo o código

Sqlite3

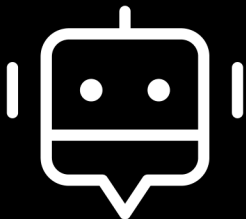
Os bancos de dados eles possuem uma divisão feito por tabelas e colunas

Nossas tabelas criadas

```
▼ Tables (3)
  > cursos
  > instrutores
  > alunos
```

Cada tabela dessas, possuem suas próprias colunas, é graças as colunas que o nosso programa consegue guardar informações, como por exemplo:

...



Lendo o código

Sqlite3

As colunas da tabela alunos

Reset Filters Records: 11					
	aluno_id		aluno_nome	aluno_curso	
	<input type="text" value="Search column..."/>		<input type="text" value="Search column..."/>	<input type="text" value="Search column..."/>	
1	65550		João	Mecânica	
2	47658		Mário	Mecânica	
3	55486		Eee	Mecânica	
4	83529		Matheus	Mecânica	
5	57944		Guilherme	Design	
6	21331		Matheus	EXCEL	
7	53969		Livia	Análise de Dados	
8	97525		Beta	Automatização	
9	53303		Thomas	Design	
10	66902		Bob	Automatização	
11	85702		Jord	Design	

Comando para puxar os dados armazenados

SELECT * FROM alunos • Todos os alunos

SELECT * FROM alunos WHERE aluno_nome = "Bob" • Todos os Bobs

SELECT aluno_curso FROM alunos • Todos os cursos de todos os alunos

Note que, o comando utiliza o nome de tabela e das colunas