

# LOGISTICS

Ciência da Computação

2o Período UNIFAGOC

professor Jacimar Tavares

QUEM SOMOS

Box Logistics

Software House

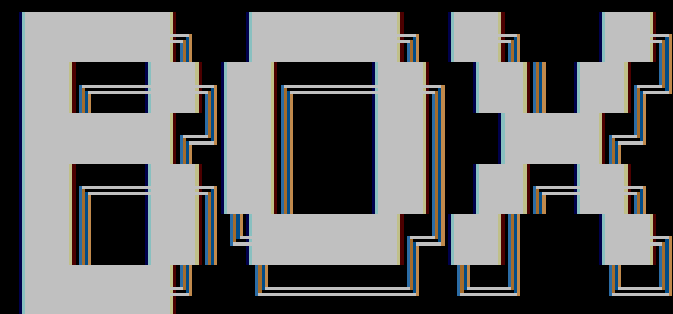


**Desenvolvemos sistemas para todas as etapas da cadeia logística**

- controle de frotas
- gestão de entregas
- relacionamento com os clientes

**GOVERNANÇA  
RASTREABILIDADE**

Conheça a  
Nossa história



BOX LOGISTICS SYSTEM ■ v0.1 (2025)  
-----  
© 2025 Box Logistics - The System

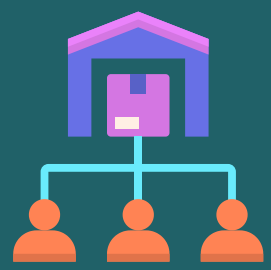
```
in: ricardo  
ha: *****  
ifique usuario e senha  
in: campo requerido  
in:
```



NAT



WE



LOGÍSTICA



GOVERNANÇA  
E COMPLIANCE



EXPERIÊNCIA  
DO USUÁRIO

Conheça  
Nosso Time



**WE**

CEO



**RIC**

TECH LEAD



**DUDA**

Customer  
Success



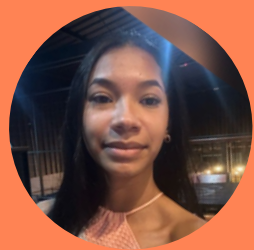
**GUII**

AGILE  
SM



**NAT**

PO



**AGA**

DESIGNER



**BIG**

DBA

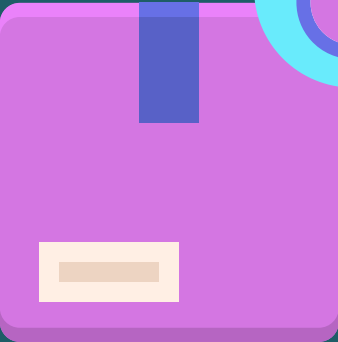


**JP**

SAC

**BOX**  
LOGISTICS

DREAM TEAM





# CADEIA

## LOGÍSTICA



compras



pagar



almoxarifado



estoque



composição



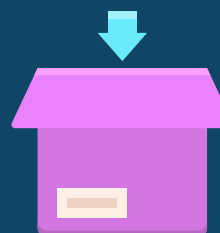
esteira



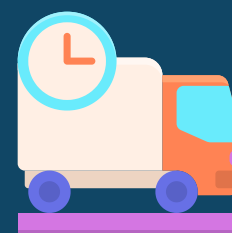
c qualidade



faturamento



separação



entregas



produtos



materiais



kanban



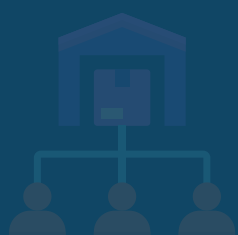
devoluções



relacionamento



delivery



armazéns



contratos



simulação



visibilidade

aguarde  
PRÓXIMAS VERSÕES



AGATTA  
DESIGNER

D:\Faculdade\Fagoc\2Periodo\untitled\fractal.exe

SISTEMA		
ID	Nome	Saldo
1	Ricardo	900
2	Alice	2500
3	Bob	-20
4	Carla	3000
ID	Nome	Saldo
1	Ricardo	900
2	Alice	2500
3	Bob	-20
4	Carla	3000

=== Render Material Layer ===

LAYER

FRACTAL

SURFACE

=== Render Aqua Layer ===

LAYER

FRACTAL

SURFACE

Configuracoes

Item 1: Configuracao

Item 2: Preferencias

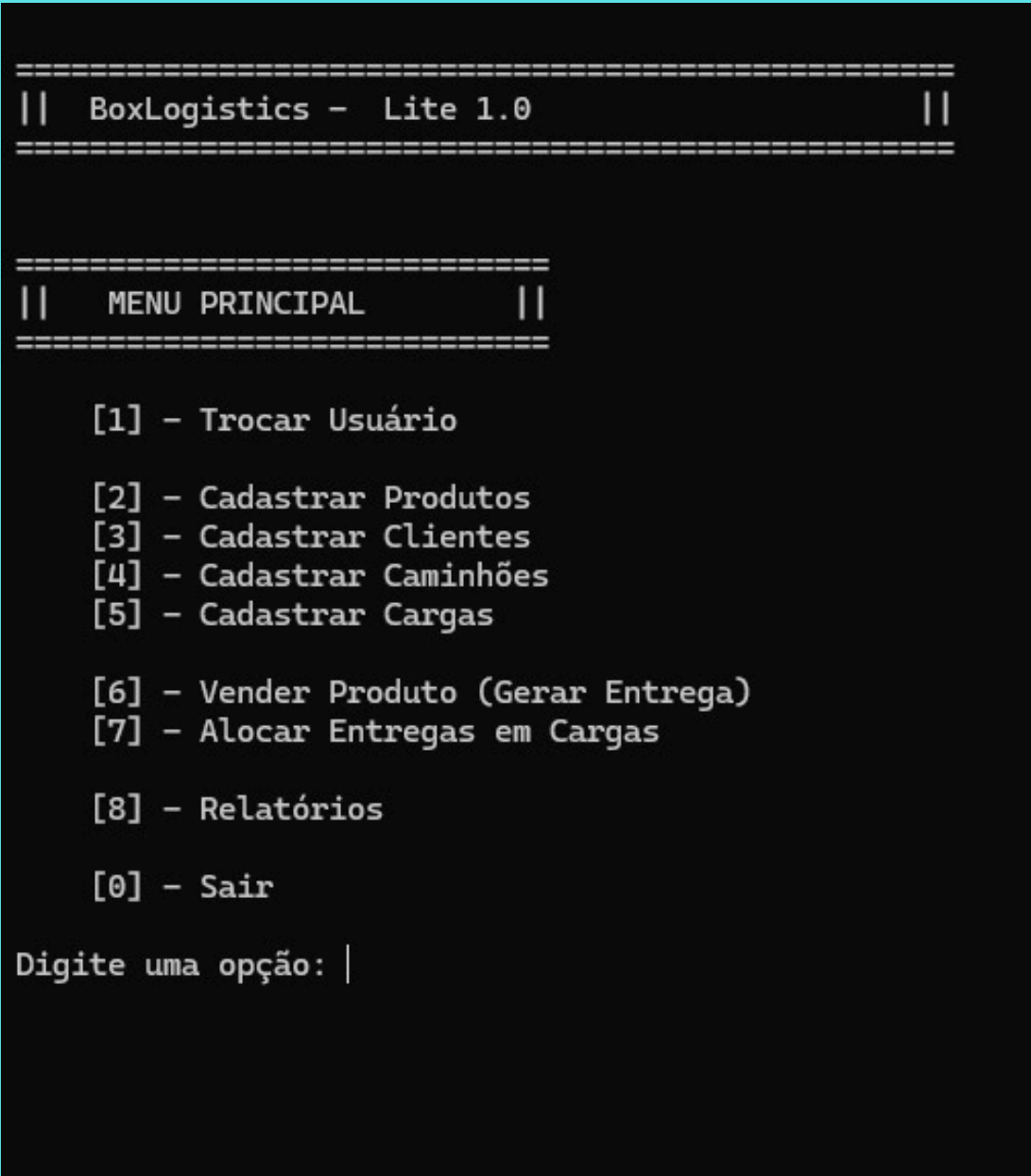
Item 3: Sistema

Item 4: Sair

[ OK ]

[ Cancelar ]

live demo



WEMERSON

CEO CRIADOR DO BOX LITE

# BOAS PRÁTICAS DE PROGRAMAÇÃO

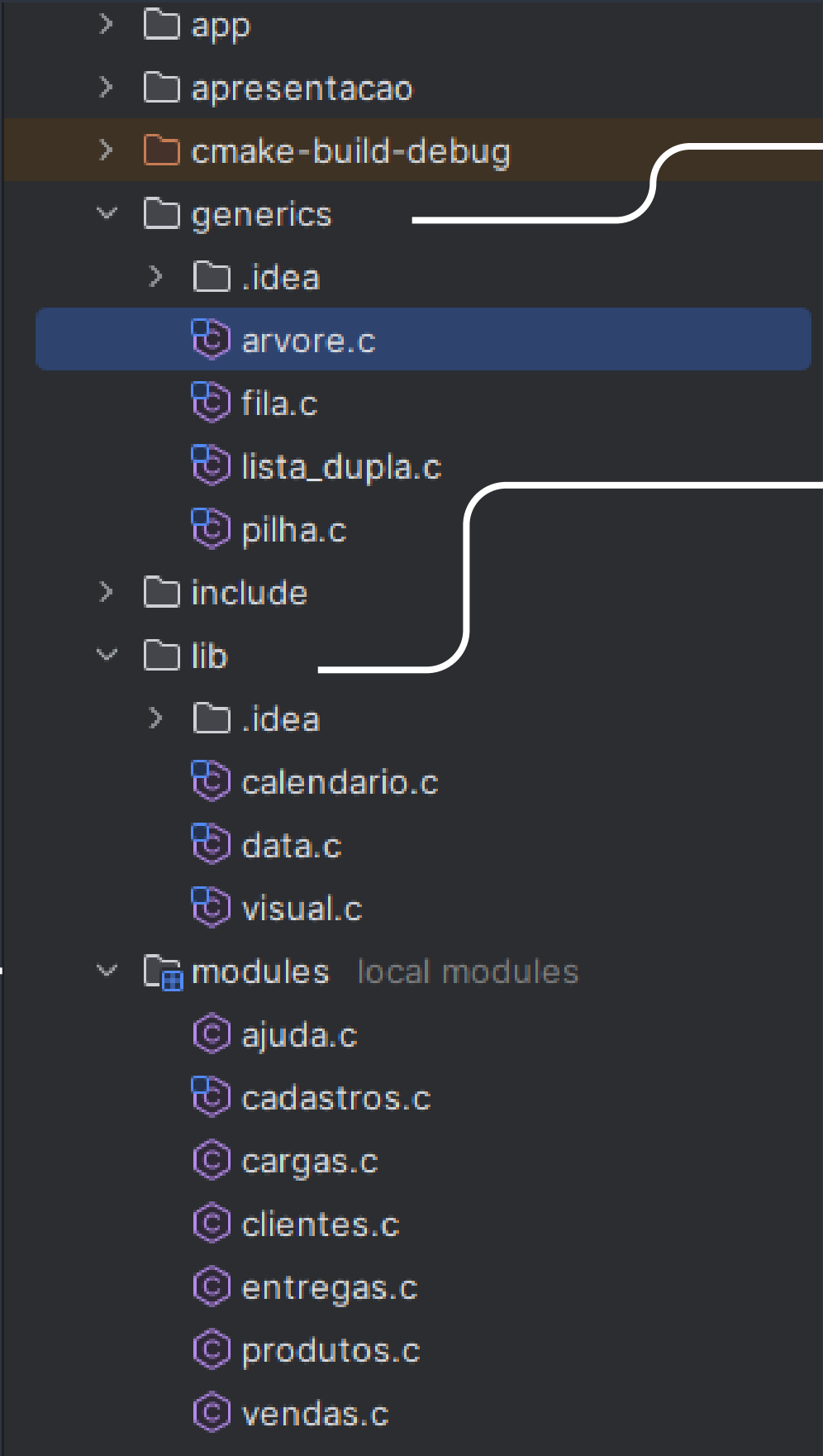
- MODULARIDADE
- SEGURANÇA

MÓDULOS



RIC

TECH LEAD



ESTRUTURAS REAPROVEITÁVEIS

BIBLIOTECAS

BOAS PRÁTICAS  
BOAS PRÁTICAS  
BOAS PRÁTICAS  
BOAS PRÁTICAS  
BOAS PRÁTICAS



# BOAS PRÁTICAS DE PROGRAMAÇÃO

```
Entrega *nova_entrega(char *destinatario, int id_produto, float valor, float quant) {  
    Entrega * result = alocar_entrega();  
    strcpy(result->destinatario, destinatario);  
    result->id_produto = id_produto;  
    result->valor = valor;  
    result->quant = quant;  
    return result;  
}
```

CONSTRUTORES

```
int validar_entrega(void* dado, char* message) {  
    Entrega *e = dado;  
    > if (e->valor <= 0) {...}  
    > if (e->quant <= 0) {...}  
    > if (e->cpf == 0) {...}  
    return VALID;  
}
```

VALIDADORES

PADRÕES DE PROJETO  
PADRÕES DE PROJETO  
PADRÕES DE PROJETO  
PADRÕES DE PROJETO  
PADRÕES DE PROJETO

# BOAS PRÁTICAS DE PROGRAMAÇÃO

```
for (int i = 0; i <= 9; i++) {
```

# ITERADOR

## ALGORITMO DE BUSCA INJEÇÃO DE DEPENDÊNCIA

[illegible]