



# Resolução de Problemas com Portugol: Cenários para a TOTVS

Neste estudo de caso, exploraremos quatro cenários desafiadores enfrentados pela TOTVS em seus diferentes setores. Para cada situação, desenvolvemos soluções utilizando a linguagem Portugol, demonstrando sua aplicação prática e capacidade de resolução de problemas reais.



# Cenário 1: RH - Otimizando o Processo de Recrutamento

## Problema

A TOTVS busca automatizar o processo de recrutamento e seleção para otimizar o tempo e reduzir custos, além de garantir maior agilidade na escolha dos melhores candidatos.

## Solução e Portugal

Utilizando algoritmos, o sistema identifica candidatos com perfil adequado às vagas, automatiza o envio de e-mails e agendamento de entrevistas, e facilita o processo de análise de currículos e testes.

# Código Portugol

## Cenário 1

```
algoritmo "Recrutamento"
var
    nome, cargo, experiencia: caractere
    idade, salarioPretendido: inteiro
    candidatoAprovado: logico
inicio
    escreva("Nome do candidato: ")
    leia(nome)
    escreva("Cargo desejado: ")
    leia(cargo)
    escreva("Experiência profissional (anos): ")
    leia(experiencia)
    escreva("Idade: ")
    leia(idade)
    escreva("Salário pretendido: ")
    leia(salarioPretendido)

    se (experiencia >= 2 e idade >= 21 e salarioPretendido <= 5000)
    entao

        candidatoAprovado <- verdadeiro
        escreva("Candidato ", nome, " aprovado para a vaga de ", cargo)
    senao
        candidatoAprovado <- falso
        escreva("Candidato ", nome, " não atende aos requisitos da vaga")
    fimse
finalgoritmo
```

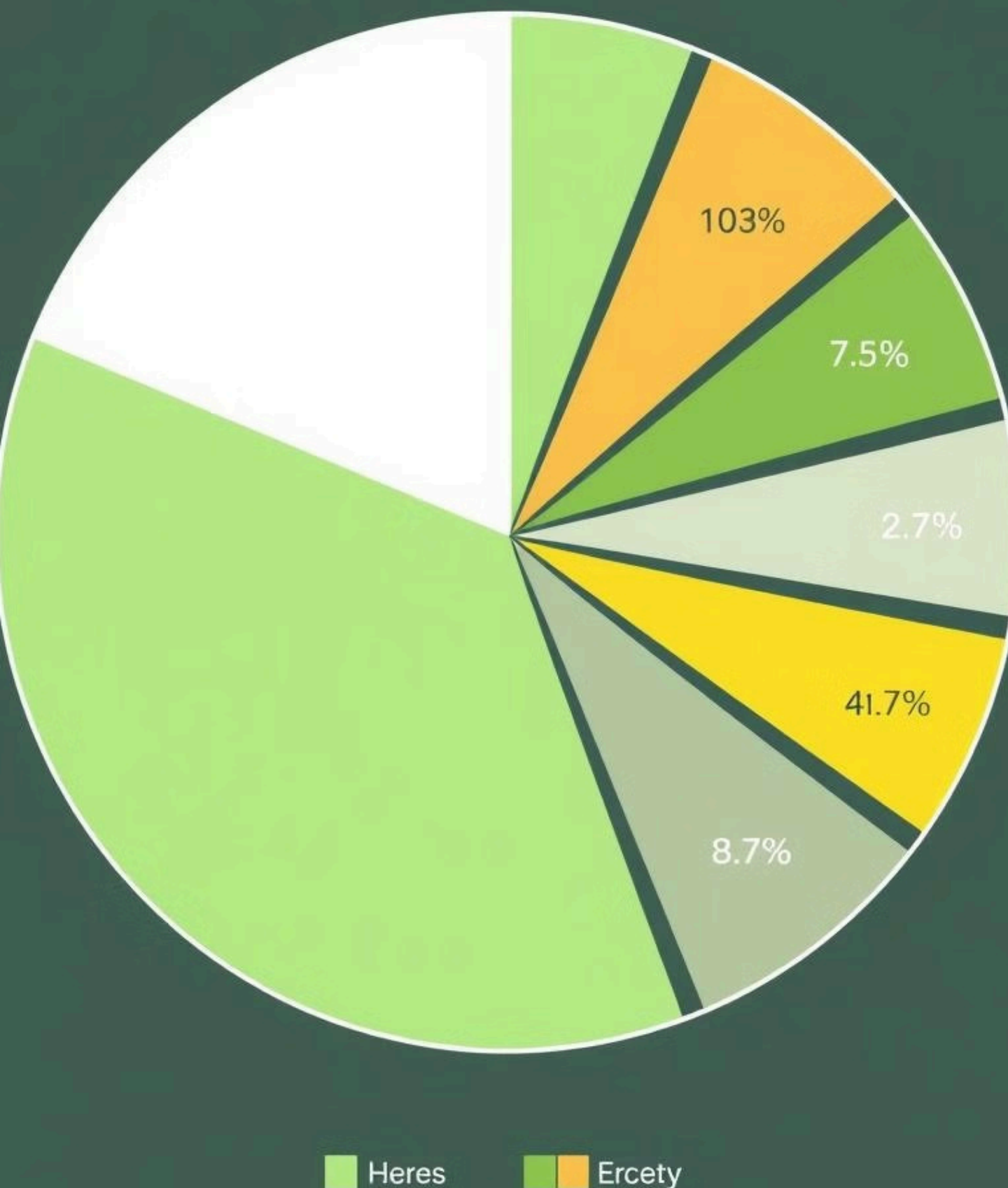


```

1 Algoritmo "semnome"
2 // Disciplina   : [Linguagem e Lógica de Programação]
3 // Professor    : Antonio Carlos Nicolodi
4 // Descrição    : Aqui você descreve o que o programa faz! (função)
5 // Autor(a)     : Nome do(a) aluno(a)
6 // Data atual   : 12/11/2024
7 var
8     nome, cargo, experiencia: caractere
9     idade, salarioPretendido: inteiro
10    candidatoAprovado: logico
11 inicio
12     escreva("Nome do candidato: ")
13     leia(nome)
14     escreva("Cargo desejado: ")
15     leia(cargo)
16     escreva("Experiência profissional (anos): ")
17     leia(experiencia)
18     escreva("Idade: ")
19     leia(idade)
20     escreva("Salário pretendido: ")
21     leia(salarioPretendido)
22
23     se (experiencia >= 2 e idade >= 21 e salarioPretendido <= 5000) entao
24         candidatoAprovado <- verdadeiro
25         escreva("Candidato ", nome, " aprovado para a vaga de ", cargo)
26     senao
27         candidatoAprovado <- falso
28         escreva("Candidato ", nome, " não atende aos requisitos da vaga")
29     fimse
30 fimalgoritmo

```

# Expense charts



## Cenário 2: Financeiro - Controlando Despesas

### Problemas

A TOTVS busca desenvolver um sistema que auxilie no controle de despesas, permitindo a análise e otimização do uso de recursos financeiros, garantindo uma gestão financeira mais eficiente.

### Solução e Portugal

O sistema implementa um algoritmo para categorizar despesas, registrar entradas e saídas, e gerar relatórios com gráficos e tabelas para análise detalhada das despesas e planejamento de ações.

# Código Portugal

## Cenário 2

```
algoritmo "ControleDespesas"
var
    descricao, categoria: caractere
    valor: real
    totalDespesas: real
inicio
    totalDespesas <- 0
    enquanto (true) faca
        escreva("Descrição da despesa: ")
        leia(descricao)
        escreva("Categoria da despesa: ")
        leia(categoria)
        escreva("Valor da despesa: ")
        leia(valor)

        totalDespesas <- totalDespesas + valor

        escreva("Deseja inserir outra despesa? (s/n)")
        se (leia(caractere) == 'n') entao

            pare
        fimse
    fimenquanto

    escreva("Total de despesas: ", totalDespesas)
finalgoritmo
```

	A	B	C	F	
	Totall	Totaled	Expense	Perfname	P
	100	270%	\$700	110,28%	
	300	390%	\$200	6,00%	
	360	270%	\$800	5,10%	
	350	380%	\$000	58.3%	
	200	380%	\$300	8,39%	
	300	130%	\$300	1,50.7	
	700	360%	\$800	5,90%	
	290	250%	\$800	9,00%	
	507	280%	\$000	5,19%	
	500	230%	2000	7,00%	
	500	257%	\$900	17,90%	
	300	250%	\$500	3,75%	
	500	380%	\$200	1,79%	
	600	500%	\$400	4,70%	
	290	200%	\$400	2,40%	
	500	500%	\$170	12,57%	
	760	280%	\$400	5,38%	
	450	980%	\$500	6,00%	
	400	570%	\$400	18,77%	

```
1 Algoritmo "semnome"
2 // Disciplina   : [Linguagem e Lógica de Programação]
3 // Professor    : Antonio Carlos Nicolodi
4 // Descrição    : Aqui você descreve o que o programa faz! (função)
5 // Autor(a)     : Nome do(a) aluno(a)
6 // Data atual   : 12/11/2024
7 var
8     descricao, categoria: caractere
9     valor: real
10    totalDespesas: real
11 inicio
12    totalDespesas <- 0
13    enquanto (true) faca
14        escreva("Descrição da despesa: ")
15        leia(descricao)
16        escreva("Categoria da despesa: ")
17        leia(categoria)
18        escreva("Valor da despesa: ")
19        leia(valor)
20
21        totalDespesas <- totalDespesas + valor
22
23        escreva("Deseja inserir outra despesa? (s/n)")
24        se (leia(caractere) == 'n') entao
25            pare
26        fimse
27    fimenquanto
28
29    escreva("Total de despesas: ", totalDespesas)
30 fimalgoritmo
```



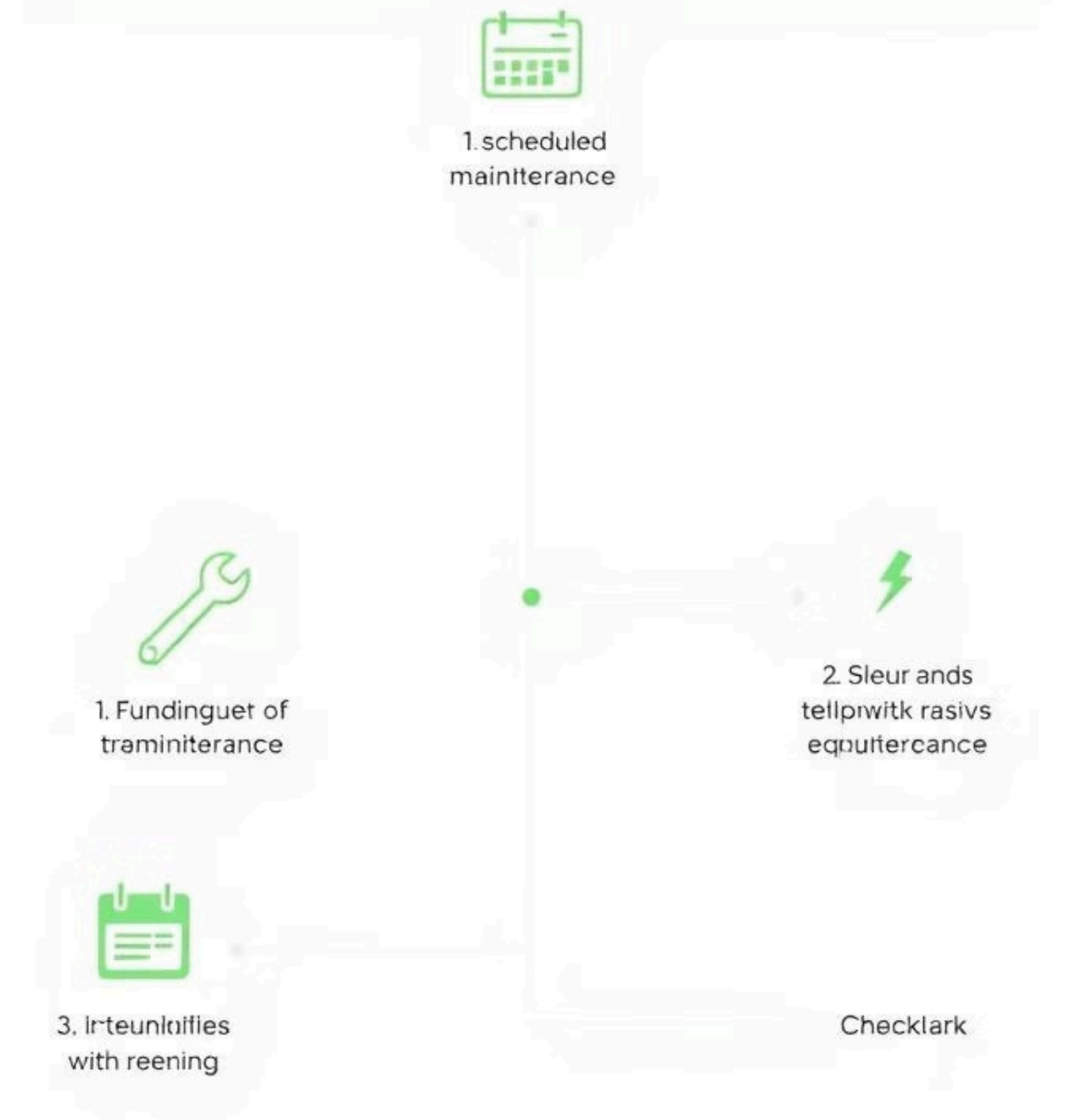
# Cenário 3: Infraestrutura

## Otimizando a Manutenção de Equipamentos

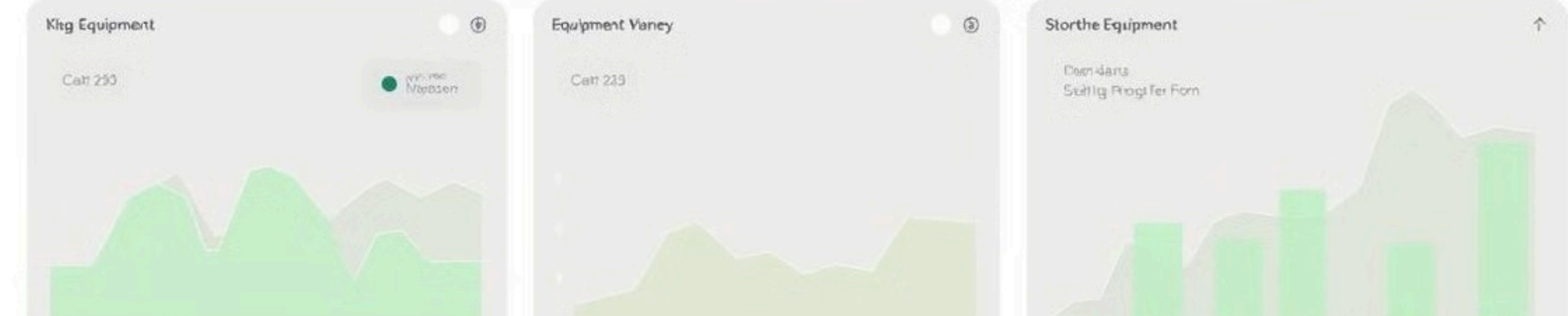
**1 Problema**  
A TOTVS busca automatizar a manutenção preventiva de seus equipamentos para garantir a operacionalidade, reduzir custos e aumentar a vida útil dos ativos.

**2 Solução e Portugal**  
O sistema implementa um algoritmo para agendar manutenções, registrar histórico de reparos e enviar alertas para serviços preventivos, garantindo a eficiência da equipe de infraestrutura.

## Equipment Maintenance







## Código Portugol - Cenário 3

```
algoritmo "ManutencaoEquipamentos"
var
nomeEquipamento, tipoManutencao: caractere
dataProximaManutencao, dataUltimoConserto: data
status: caractere
inicio
    escreva("Nome do equipamento: ")
    leia(nomeEquipamento)
    escreva("Tipo de manutenção: ")
    leia(tipoManutencao)
    escreva("Data da última manutenção: ")
    leia(dataUltimoConserto)

    se (tipoManutencao == "preventiva") entao

        escreva("Data da próxima manutenção: ")
        leia(dataProximaManutencao)
        se (dataProximaManutencao < dataAtual) entao
            escreva("Alerta: Manutenção preventiva atrasada!")
            status <- "Atrasada"
        senao
            status <- "Em dia"
        fimse
    senao
        status <- "Sem necessidade de manutenção"
    fimse
    escreva("Status da manutenção: ", status)
finalgoritmo
```

```
1 Algoritmo "semnome"
2 // Disciplina   : [Linguagem e Lógica de Programação]
3 // Professor    : Antonio Carlos Nicolodi
4 // Descrição    : Aqui você descreve o que o programa faz! (função)
5 // Autor(a)     : Nome do(a) aluno(a)
6 // Data atual   : 12/11/2024
7 var
8     nomeEquipamento, tipoManutencao: caractere
9     dataProximaManutencao, dataUltimoConserto: data
10    status: caractere
11 inicio
12     escreva("Nome do equipamento: ")
13     leia(nomeEquipamento)
14     escreva("Tipo de manutenção: ")
15     leia(tipoManutencao)
16     escreva("Data da última manutenção: ")
17     leia(dataUltimoConserto)
18
19     se (tipoManutencao == "preventiva") entao
20         escreva("Data da próxima manutenção: ")
21         leia(dataProximaManutencao)
22         se (dataProximaManutencao < dataAtual) entao
23             escreva("Alerta: Manutenção preventiva atrasada!")
24             status <- "Atrasada"
25         senao
26             status <- "Em dia"
27         fimse
28     senao
29         status <- "Sem necessidade de manutenção"
30     fimse
31
32     escreva("Status da manutenção: ", status)
33 fimalgoritmo
```

## Marketing canposins



# Cenário 4: Marketing Analisando Dados de Campanhas



## Problema

A TOTVS busca analisar os dados de suas campanhas de marketing para otimizar seus investimentos, entender o público-alvo e criar campanhas mais eficazes.



## Solução e Portugal

O sistema implementa um algoritmo para coletar dados de diversas fontes, como anúncios online, e-mail marketing e redes sociais, e gerar relatórios personalizados com insights sobre o desempenho das campanhas.

	Friendly Click	Play Click
	Marketide renpage	Friem Inves
s	9,71	1,7
npagens	1,316	9,5
s		
click	1,80	1,2
r empaims	1,78	2,1
ck	1093	1,2
ck	1890	1,2
ting	1016	1
npaigns	1,355	3,5
mplye	984	
ck	1,377	2,7
vestmant	996	1,4
	11,010	124

# Código Portugal

## Cenário 4

```

algoritmo "AnaliseCampanhas"
var
nomeCampanha, tipoCampanha: caractere
clicks, conversao, custoPorClick, retornoInvestimento: real
inicio
  escreva("Nome da campanha: ")
  leia(nomeCampanha)
  escreva("Tipo da campanha: ")
  leia(tipoCampanha)
  escreva("Número de cliques: ")
  leia(clicks)
  escreva("Taxa de conversão: ")
  leia(conversao)
  escreva("Custo por clique: ")
  leia(custoPorClick)

  retornoInvestimento <- (conversao * 100) - (clicks * custoPorClick)

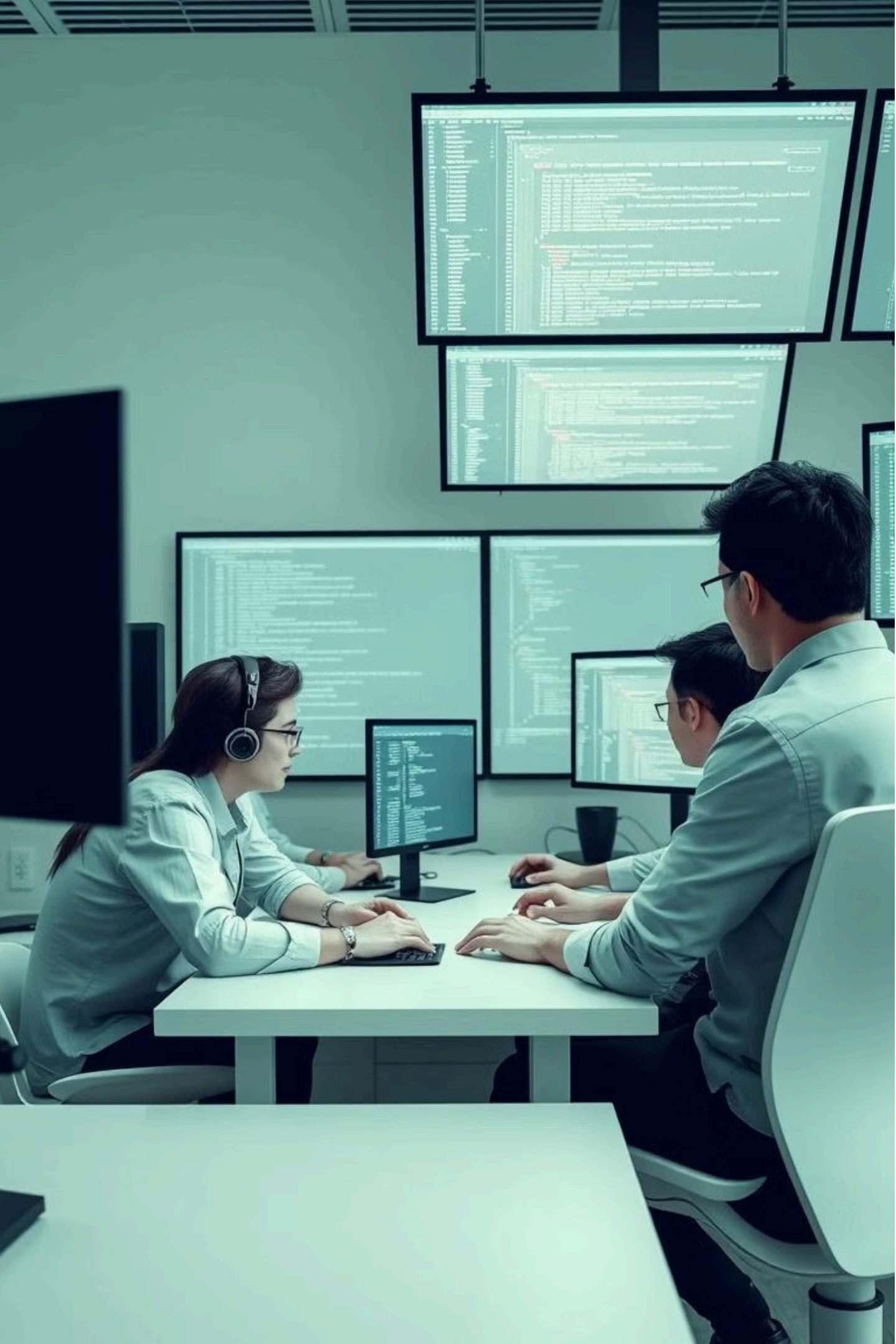
  escreva("Retorno sobre o investimento: ", retornoInvestimento)
finalgoritmo

```



## Área dos algoritmos ( Edição do código fonte ) -> Nome do arquivo: [semnome]

```
1 Algoritmo "semnome"
2 // Disciplina      : [Linguagem e Lógica de Programação]
3 // Professor       : Antonio Carlos Nicolodi
4 // Descrição       : Aqui você descreve o que o programa faz! (função)
5 // Autor(a)        : Nome do(a) aluno(a)
6 // Data atual      : 12/11/2024
7 var
8     nomeCampanha, tipoCampanha: caractere
9     clicks, conversao, custoPorClick, retornoInvestimento: real
10 inicio
11     escreva("Nome da campanha: ")
12     leia(nomeCampanha)
13     escreva("Tipo da campanha: ")
14     leia(tipoCampanha)
15     escreva("Número de cliques: ")
16     leia(clicks)
17     escreva("Taxa de conversão: ")
18     leia(conversao)
19     escreva("Custo por clique: ")
20     leia(custoPorClick)
21
22     retornoInvestimento <- (conversao * 100) - (clicks * custoPorClick)
23
24     escreva("Retorno sobre o investimento: ", retornoInvestimento)
25 fimalgoritmo
```



# Conclusão: Portugol como Ferramenta Essencial

Utilizando Portugol, podemos desenvolver soluções eficazes para os desafios enfrentados pela TOTVS em seus diferentes setores. A linguagem facilita a criação de algoritmos eficientes e legíveis, promovendo a compreensão e a colaboração entre desenvolvedores.