SESI SENAI

Técnico em

DESENVOLVIMENTO DE SISTEMAS

Técnico em
DESENVOLVIMENTO DE SISTEMAS

SESI SENAI

AULA 3

UNIDADE CURRICULAR

PROGRAMAÇÃO BACK-END

Técnico em
DESENVOLVIMENTO DE SISTEMAS

SESI SENAI

UNIDADE CURRICULAR
PROGRAMAÇÃO BACK-END

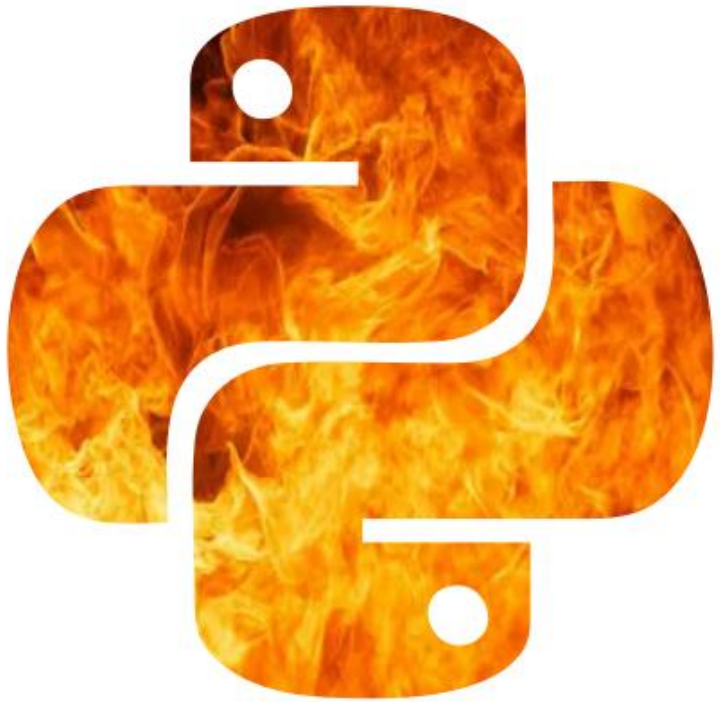
Programação em PYTHON

NO EPISÓDIO ANTERIOR...

Técnico em
DESENVOLVIMENTO DE SISTEMAS

SESI SENAI

UNIDADE CURRICULAR
PROGRAMAÇÃO BACK-END



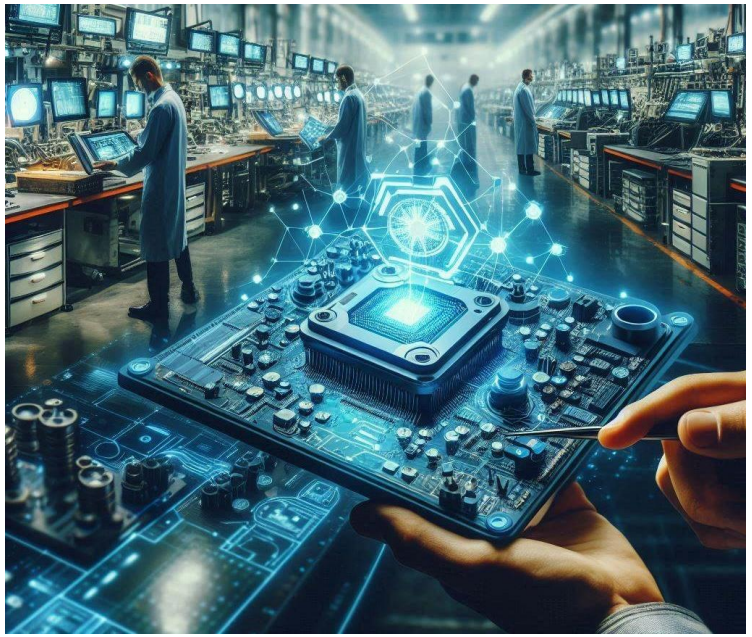
AQUECIMENTO PYTHON

CONTEXTO

AQUECIMENTO PYTHON

TECNOZ

A empresa **Tecnoz** é especializada na produção de componentes eletrônicos de alta precisão. Com uma demanda crescente no mercado, a qualidade dos produtos fabricados é crucial para manter a confiança dos clientes e garantir contratos futuros. Para assegurar a excelência, a Tecnoz contratou um time de desenvolvedores de software para criar novas ferramentas que ajudarão a indústria a elevar ainda mais a sua qualidade.



Você foi contratado(a) como programador(a) para no time de desenvolvedores da Tecnoz.

Técnico em
DESENVOLVIMENTO DE SISTEMAS

SESI SENAI

UNIDADE CURRICULAR
PROGRAMAÇÃO BACK-END

SITUAÇÃO-PROBLEMA

1

Aquecimento PYTHON

TECNOZ

Desafio:
Você deverá desenvolver uma ferramenta que ajudará o departamento de controle de qualidade da Tecnoz. Seu programa deverá solicitar ao operador a quantidade total de produtos fabricados em um lote e a quantidade de produtos identificados como defeituosos. O programa deverá calcular a porcentagem de defeitos nesse lote e exibir o resultado.

Além de calcular a porcentagem de defeitos, o programa deverá:

1. Verificar se a porcentagem de defeitos está dentro do limite aceitável de 5%.
2. Caso a porcentagem seja superior ao limite, o programa deverá sugerir que o lote seja reavaliado antes de ser enviado ao cliente.
3. O programa também deverá exibir uma mensagem adicional que indique se o lote está "Aprovado para envio" ou "Reprovado - Reavaliar".

Exemplo de saída do programa:
•Total de produtos fabricados: 1000
•Produtos defeituosos: 60
•Porcentagem de defeitos: 6%
•Status: Reprovado – Reavaliar

Este programa será uma peça chave para a manutenção dos padrões de qualidade da TecnoIndústria, garantindo que apenas produtos de alta qualidade cheguem ao mercado.

Técnico em
DESENVOLVIMENTO DE SISTEMAS

SESI SENAI

UNIDADE CURRICULAR
PROGRAMAÇÃO BACK-END

SITUAÇÃO-PROBLEMA

2

AQUECIMENTO PYTHON

TECNOZ

Desafio:

Você foi encarregado(a) de desenvolver um programa que ajude a Tecnoz a monitorar e otimizar a eficiência energética das máquinas na linha de produção. O programa deverá solicitar ao operador o consumo de energia de uma máquina em kilowatts (kWh) e o número de peças produzidas durante um determinado período.

Além de calcular a eficiência energética básica, o programa deverá:

1. Calcular a eficiência energética da máquina em termos de peças por kilowatt-hora (peças/kWh).
2. Se a eficiência for menor que 10 peças por kWh, o programa deverá exibir uma mensagem de alerta indicando baixa eficiência, sugerindo uma revisão do processo ou manutenção da máquina.
3. O programa deverá permitir que o operador insira uma meta de eficiência energética (por exemplo, 12 peças/kWh), comparando o resultado atual com a meta e exibindo uma mensagem que indique se a meta foi atingida ou não.
4. Se a eficiência estiver abaixo de 5 peças por kWh, o programa deverá emitir um alerta crítico e recomendar a parada imediata da máquina para inspeção.

Exemplo de saída do programa:

•Consumo de energia: 500 kWh
•Peças produzidas: 4000
•Eficiência energética: 8 peças/kWh
•Status: Baixa eficiência - Revisar máquina
•Meta de eficiência: 12 peças/kWh
•Alerta: Meta não atingida - Considere ajustes na operação

Este programa será essencial para a Tecnoz, ajudando a garantir que as máquinas operem com a máxima eficiência energética, reduzindo custos e promovendo práticas sustentáveis na produção de componentes eletrônicos.

Técnico em
DESENVOLVIMENTO DE SISTEMAS

SESI SENAI

UNIDADE CURRICULAR
PROGRAMAÇÃO BACK-END

SITUAÇÃO-PROBLEMA

3

Aquecimento PYTHON

TECNOZ

Desafio:

Você recebeu a tarefa de criar um programa que ajude a monitorar a produtividade da linha de montagem da Tecnoz. O programa deverá solicitar ao operador o número de peças montadas por hora e o tempo total de operação da linha de montagem em um dia específico. Além de calcular a produção total diária, o programa deverá:

1. Calcular a produção total com base no número de peças montadas por hora e no tempo total de operação.
2. Verificar se a produção atingiu ou superou a meta diária de 1.000 peças.
3. Se a produção estiver abaixo da meta, o programa deverá calcular quantas horas adicionais seriam necessárias para atingir a meta, assumindo a mesma taxa de produção por hora.
4. Implementar uma funcionalidade para que o operador insira uma meta personalizada de produção para dias específicos (por exemplo, em dias de alta demanda), permitindo comparação com a meta padrão de 1.000 peças.
5. Se a produção estiver abaixo de 800 peças, o programa deverá emitir um alerta de baixa produtividade, sugerindo que ajustes na linha de montagem ou revisões de turnos sejam considerados.

Exemplo de saída do programa:

- Peças montadas por hora: 80
- Tempo total de operação: 10 horas
- Produção total: 800 peças
- Status: Produção abaixo da meta - Faltam 200 peças
- Horas adicionais necessárias para atingir a meta de 1.000 peças: 2.5 horas
- Alerta: Baixa produtividade - Revisar operação
- Meta personalizada: 1.200 peças

Este programa será uma ferramenta crucial para a Tecnoz, permitindo que a equipe de produção monitore e ajuste as operações da linha de montagem em tempo real, garantindo que as metas de produção sejam consistentemente atingidas e otimizadas.

Técnico em
DESENVOLVIMENTO DE SISTEMAS

SESI SENAI

UNIDADE CURRICULAR
PROGRAMAÇÃO BACK-END

SITUAÇÃO-PROBLEMA

4

Aquecimento PYTHON

TECNOZ

Desafio:
Você foi encarregado(a) de desenvolver um programa que ajude a calcular e analisar os custos de produção de forma detalhada. O programa deverá solicitar ao usuário os custos de matéria-prima e mão de obra, além do número total de unidades produzidas em um lote.

Além de calcular o custo unitário de produção, o programa deverá:

1. Calcular o custo unitário de produção com base no custo total da matéria-prima e da mão de obra dividido pelo número de unidades produzidas.
2. Permitir que o usuário defina um valor de referência para o custo unitário, com base nos padrões da indústria ou metas internas da empresa.
3. Informar se o custo unitário calculado está acima, abaixo ou dentro do valor de referência definido pelo usuário.
4. Se o custo unitário estiver acima do valor de referência, o programa deverá sugerir possíveis áreas de otimização, como reduzir o custo da matéria-prima ou melhorar a eficiência da mão de obra.

Exemplo de saída do programa:
Custo de matéria-prima: R$ 50.000,00
Custo de mão de obra: R$ 30.000,00
Unidades produzidas: 10.000
Custo unitário de produção: R$ 8,00 por unidade
Valor de referência: R$ 7,50 por unidade
Status: Custo acima do valor de referência
Sugestão: Avaliar fornecedores de matéria-prima ou otimizar processos de produção

Este programa será essencial para a Tecnoz, ajudando a monitorar e controlar os custos de produção de forma eficiente, além de proporcionar insights valiosos para a tomada de decisões estratégicas que possam melhorar a rentabilidade da empresa.

Análise e interpretação do problema

- "Um problema bem definido é um problema metade resolvido."

Charles Kettering, um renomado inventor e engenheiro americano.

É fundamental identificar as entradas e o que se espera como resultado do programa/algoritmo.

A interpretação de texto, pensamento crítico e visão analítica são habilidades fundamentais para qualquer profissão, inclusive para desenvolvedor de softwares.

NO EPISÓDIO DE HOJE...

Técnico em
DESENVOLVIMENTO DE SISTEMAS

SESI SENAI

UNIDADE CURRICULAR
PROGRAMAÇÃO
BACK-END

FUNÇÃO

FUNÇÃO

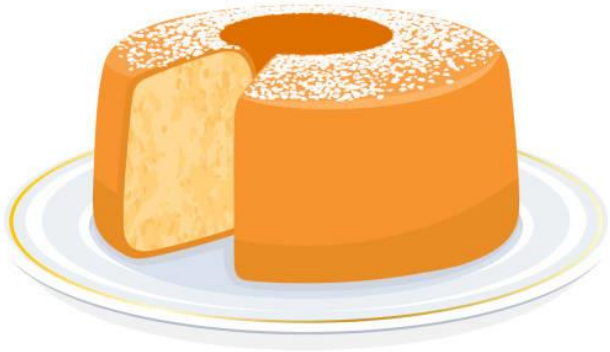
Na programação, **funções** são blocos de código que realizam determinadas tarefas que normalmente precisam ser executadas diversas vezes dentro de uma aplicação.

Quando surge essa necessidade, para que várias instruções não precisem ser repetidas, elas são agrupadas em uma função, à qual é dado um nome e que poderá ser chamada/executada em diferentes partes do programa.

EXEMPLO: *Algoritmo – Fazer um bolo*

Ingredientes:

- 2 xícaras (chá) farinha de trigo
- 2 xícaras (chá) açúcar
- 1 xícara (chá) leite
- 2 ovos
- 4 colheres (sopa) óleo
- 1 pitada de sal
- 1 colher (sopa) fermento em pó



Modo de Preparo:

1. Na batedeira, bata os ingredientes por aproximadamente 5 minutos.
2. Retire a tigela da batedeira;
3. Coloque o fermento e misture cuidadosamente na massa.
4. Despeje a massa do bolo em uma forma untada.
5. Leve ao forno preaquecido a 180ºC por aproximadamente 35 minutos.

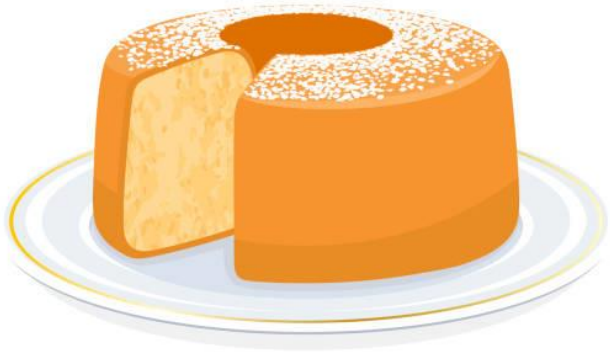
python™

FUNÇÃO

EXEMPLO: *Algoritmo – Fazer um bolo*

Ingredientes:

- 2 xícaras (chá) farinha de trigo
- 2 xícaras (chá) açúcar
- 1 xícara (chá) leite
- 2 ovos
- 4 colheres (sopa) óleo
- 1 pitada de sal
- 1 colher (sopa) fermento em pó



Modo de Preparo:

1. Na batedeira, bata os ingredientes por aproximadamente 5 minutos.
2. Retire a tigela da batedeira;
3. Coloque o fermento e misture cuidadosamente na massa.
4. Despeje a massa do bolo em uma forma untada.
5. Leve ao forno preaquecido a 180ºC por aproximadamente 35 minutos.

python™

FUNÇÃO

FUNÇÃO

EXEMPLO: *Algoritmo – Fazer um bolo*

Função: Levar_ao_forno(Tempo: ___; Temperatura: ____)

Algoritmo da função:

1. Vá até o fogão;
2. Pré aqueça o forno;
3. Abra o forno;
4. Coloque a forma;
5. Feche o forno;
6. Deixe assando pelo tempo e temperatura indicados;
7. Desligue o forno quando o tempo terminar;
8. Abra o forno;
9. Retire a forma;
10. Feche o forno;

Fim.

FUNÇÃO

EXEMPLO: *Algoritmo – Fazer um bolo*

Função: Levar_ao_forno(Tempo: ___; Temperatura: ____)

Algoritmo da função:

1. Vá até o fogão;
2. **Pre_aquecer_forno( )** **Outra função**
3. Abra o forno;
4. Coloque a forma;
5. Feche o forno;
6. Deixe assando pelo tempo e temperatura indicados;
7. Desligue o forno quando o tempo terminar;
8. Abra o forno;
9. Retire a forma;
10. Feche o forno;

Fim.

EXEMPLO: *Algoritmo – Fazer um bolo*

python™

FUNÇÃO

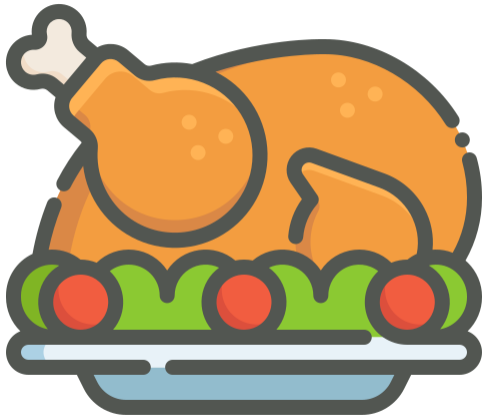
Função: Pre_aquecer_forno( )

Algoritmo da função:

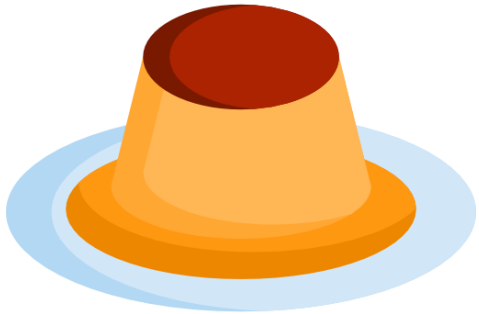
1. Vá até o fogão;
2. Acenda o forno;
3. Coloque em 180° C;
4. Deixe aquecer por 10 minutos;

Fim.

EXEMPLO: *Outras receitas*



Levar_ao_forno(120m; 160°C)



Levar_ao_forno(90m; 200°C)

python™

FUNÇÃO

Um função pode ser reaproveitada em outras ações.

FUNÇÃO EM PYTHON:

A sintaxe de uma função é definida por três partes: **nome**, **parâmetros** e **corpo**, o qual agrupa uma sequência de linhas que representa algum comportamento. No código abaixo, temos um exemplo de declaração de função em Python:

```
def nome_funcao(parametros):
        #códigos da função
```

FUNÇÃO

FUNÇÃO

CRIANDO A FUNÇÃO EM PYTHON:

```
def boasvindas(nome_usuario):
        print(f“Seja bem-vindo(a) {nome_usuario}”)
```

CHAMANDO A FUNÇÃO EM PYTHON:

```
boasvindas(“Marina”)
```

FUNÇÃO SEM ARGUMENTOS

CRIANDO A FUNÇÃO EM PYTHON:

```
def imprimir_linha():
        print(" - - - - - - - - - - - - - - - - - - - - - - - - - - - ")
```

CHAMANDO A FUNÇÃO EM PYTHON:

```
imprimir_linha()
```

python™

FUNÇÃO

FUNÇÃO COM RETORNO DE DADOS:

FUNÇÃO

```
def soma(num1, num2):
        soma = valor1 + valor2
        return soma
```

CHAMANDO A FUNÇÃO EM PYTHON:

```
print(soma(12,10))
```

CRIANDO FUNÇÕES

FUNÇÃO

Crie um programa que contenha quatro funções, uma para cada operação matemática (soma, subtração, multiplicação, e divisão). Chame as funções no código.

Técnico em
DESENVOLVIMENTO DE SISTEMAS

SESI SENAI

UNIDADE CURRICULAR
PROGRAMAÇÃO
BACK-END

CRIANDO FUNÇÕES

CALCULADORA

python™

FUNÇÃO

Crie um programa que contenha uma função calculadora, onde o usuário insere qual operação deseja fazer e dois números.

FUNÇÃO sem retorno e sem argumento

```
def imprimir( ):
        print(“- - - - - - -”)
```

FUNÇÃO sem retorno e com argumento

```
def imprimir(quant):
    for x in range(quant):
        print(“- - - - - - -”)
```

FUNÇÃO com retorno e sem argumento

```
def calc_data( ):
        return 2024 - 1950
```

FUNÇÃO com retorno e com argumento

```
def calc_data(ano):
        return 2024 - ano
```

FUNÇÃO

Exercícios com funções em Python

SITUAÇÃO-PROBLEMA

python™

FUNÇÃO

1 – CONVERSOR DE TEMPERATURAS

Crie um programa que contenha funções para converter entre diferentes unidades de temperatura, como Celsius para Fahrenheit e vice-versa, Celsius pra Kelvin, Fahrenheit para Kelvin e vice-versa.
Peça ao usuário escolher como deseja converter (de qual unidade para qual unidade de temperatura) depois o valor da temperatura que deseja converter. Mostre o resultado.

SITUAÇÃO-PROBLEMA

python™

FUNÇÃO

2 - É Primo ou não é?

Escreva um código que contenha uma função que verifique se um número informado pelo usuário é Primo ou não. Retorne ***True*** ou ***False*** para o programa principal.

O que é número primo?
Um número primo é aquele que é dividido apenas por um e por ele mesmo.

SITUAÇÃO-PROBLEMA

python™

FUNÇÃO

3 – CONVERSOR DE DISTÂNCIAS

Alguns países usam escalas diferentes para medir distâncias.

Para resolver, crie um programa com funções que convertam distâncias:

- Quilômetros para milhas e vice-versa;
- Metros para jardas e vice-versa;
- Centímetros para polegadas e vice-versa;

PLUS: Adicione outras escalas!

SITUAÇÃO-PROBLEMA

python™

FUNÇÃO

4 – CÁLCULO DE JUROS COMPOSTOS

Crie um programa que contenha uma função para cálculo de juros compostos.
O usuário deverá inserir o valor inicial investido, a taxa de juros em porcentagem (a.m.%) e a quantidade de meses que deseja investir.

Use a função para calcular e mostrar o resultado final.