

Pseudocódigo

```
Função parse_time(tempo: string) -> inteiro:  
    Dividir tempo em horas e minutos usando ':' como separado  
r.
```

```
    Retornar (horas * 60) + minutos.
```

```
Função format_time(minutos: inteiro) -> string:
```

```
    horas ← minutos // 60
```

```
    minutos_restantes ← minutos % 60
```

```
    Retornar string formatada como "HH:MM".
```

```
Função find_longest_nap(agenda: lista de tuplas) -> (inteiro,  
inteiro):
```

```
    inicio_expediente ← parse_time("10:00")
```

```
    fim_expediente ← parse_time("18:00")
```

```
    Adicionar (inicio_expediente, inicio_expediente) à agend  
a.
```

```
    Adicionar (fim_expediente, fim_expediente) à agenda.
```

```
    Ordenar agenda por horário de início.
```

```
    max_nap ← 0
```

```
    nap_start ← inicio_expediente
```

```
    Para i de 1 até tamanho da agenda:
```

```
        fim_anterior ← agenda[i-1][1]
```

```
        inicio_atual ← agenda[i][0]
```

```
        intervalo_livre ← inicio_atual - fim_anterior
```

```

    Se intervalo_livre > max_nap:
        max_nap ← intervalo_livre
        nap_start ← fim_anterior

Retornar (nap_start, max_nap)

Função process_file(caminho_arquivo: string) -> lista de strings:
    Ler todas as linhas do arquivo.
    resultados ← lista vazia
    i ← 0

    Enquanto i < total de linhas:
        n ← converter linhas[i] para inteiro
        i ← i + 1

        agenda ← lista vazia

        Para cada evento de 1 até n:
            tempo_inicial, tempo_final ← extrair horários da
linha[i]
            inicio ← parse_time(tempo_inicial)
            fim ← parse_time(tempo_final)
            Adicionar (inicio, fim) à agenda.
            i ← i + 1

        (nap_start, nap_duration) ← find_longest_nap(agenda)

        nap_start_formatado ← format_time(nap_start)

    Se nap_duration >= 60:
        horas ← nap_duration // 60
        minutos ← nap_duration % 60
        Adicionar ao resultados: "O cochilo mais longo co
meça às {nap_start_formatado} e dura {horas} horas e {minuto
s} minutos."

```

```
Caso contrário:
    Adicionar ao resultados: "O cochilo mais longo co
meça às {nap_start_formatado} e dura {nap_duration} minutos."

Retornar resultados.

Função main():
    arquivo ← "input.txt"
    resultados ← process_file(arquivo)
    Para cada resultado em resultados:
        Imprimir resultado.
```

parse_time e format_time: Convertem entre formatos HH:MM e minutos para facilitar cálculos e exibir resultados.

find_longest_nap: Encontra o maior intervalo livre na agenda após adicionar os limites do expediente e ordenar.

process_file: Processa a entrada do arquivo, calcula o maior intervalo livre para cada dia e formata os resultados.

main: Lida com a entrada e exibe os resultados.